

|                |     |              |                  |
|----------------|-----|--------------|------------------|
| DDDDDDDDDDDDDD |     | CCCCCCCCCCCC | LLL              |
| DDDDDDDDDDDDDD |     | CCCCCCCCCCCC | LLL              |
| DDDDDDDDDDDDDD |     | CCCCCCCCCCCC | LLL              |
| DDD            | DDD | CCC          | LLL              |
| DDD            | DDD | CCC          | LLL              |
| DDD            | DDD | CCC          | LLL              |
| DDD            | DDD | CCC          | LLL              |
| DDD            | DDD | CCC          | LLL              |
| DDD            | DDD | CCC          | LLL              |
| DDD            | DDD | CCC          | LLL              |
| DDD            | DDD | CCC          | LLL              |
| DDD            | DDD | CCC          | LLL              |
| DDD            | DDD | CCC          | LLL              |
| DDD            | DDD | CCC          | LLL              |
| DDD            | DDD | CCC          | LLL              |
| DDD            | DDD | CCC          | LLL              |
| DDDDDDDDDDDDDD |     | CCCCCCCCCCCC | LLLLLLLLLLLLLLLL |
| DDDDDDDDDDDDDD |     | CCCCCCCCCCCC | LLLLLLLLLLLLLLLL |
| DDDDDDDDDDDDDD |     | CCCCCCCCCCCC | LLLLLLLLLLLLLLLL |

```
RRRRRRRR      EEEEEEEEEE      AAAAAA      DDDDDDDD      RRRRRRRR      EEEEEEEEEE      CCCCCCCC
RRRRRRRR      EEEEEEEEEE      AAAAAA      DDDDDDDD      RRRRRRRR      EEEEEEEEEE      CCCCCCCC
RR      RR      EE      AA      AA      DD      DD      RR      RR      EE      CC
RR      RR      EE      AA      AA      DD      DD      RR      RR      EE      CC
RR      RR      EE      AA      AA      DD      DD      RR      RR      EE      CC
RR      RR      EE      AA      AA      DD      DD      RRRRRRRR      EEEEEEEE      CC
RRRRRRRR      EEEEEEEEEE      AA      AA      DD      DD      RRRRRRRR      EEEEEEEE      CC
RRRRRRRR      EEEEEEEEEE      AA      AA      DD      DD      RR      RR      EE      CC
RR      RR      EE      AAAAAAAAAA      DD      DD      RR      RR      EE      CC
RR      RR      EE      AAAAAAAAAA      DD      DD      RR      RR      EE      CC
RR      RR      EE      AA      AA      DD      DD      RR      RR      EE      CC
RR      RR      EE      AA      AA      DD      DD      RR      RR      EE      CC
RR      RR      EEEEEEEEEE      AA      AA      DDDDDDDD      RR      RR      EEEEEEEEEE      CCCCCCCC
RR      RR      EEEEEEEEEE      AA      AA      DDDDDDDD      RR      RR      EEEEEEEEEE      CCCCCCCC
```

```
LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLL      IIIIII      SSSSSSSS
```

|      |      |   |
|------|------|---|
| (3)  | 160  | READ NEXT INPUT RECORD                    |
| (8)  | 501  | PROCESS RECALL COMMANDS                   |
| (11) | 688  | PROCESS ESCAPE SEQUENCES                  |
| (12) | 887  | RESTORE LOCKED KEYPAD STATE               |
| (13) | 919  | EXPAND INPUT LINE                         |
| (14) | 1072 | SPECIAL TOKEN LEXICAL PROCESSING          |
| (15) | 1099 | PROCESS &SYMBOL CONSTRUCT                 |
| (16) | 1130 | PROCESS &FILESPEC CONSTRUCT               |
| (17) | 1158 | ERROR HANDLER IN CHARACTER INPUT ROUTINES |
| (18) | 1180 | RECALL COMMAND                            |

```

0000 1 .TITLE READREC - READ AN INPUT RECORD
0000 2 .IDENT 'V04-000'
0000 3 .DEFAULT DISPLACEMENT,WORD
0000 4
0000 5
0000 6 *****
0000 7 *
0000 8 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 9 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 10 * ALL RIGHTS RESERVED.
0000 11 *
0000 12 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 13 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 14 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 15 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 16 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 17 * TRANSFERRED.
0000 18 *
0000 19 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 20 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 21 * CORPORATION.
0000 22 *
0000 23 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 24 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 25 *
0000 26 *
0000 27 *****
0000 28
0000 29 ABSTRACT:
0000 30
0000 31 These routines are called by the lexical processing routines to
0000 32 perform functions which are optional to the basic lexical routines,
0000 33 but are required by CLI parsing.
0000 34
0000 35 AUTHOR:
0000 36
0000 37 Tim Halvorsen, Jan-1982
0000 38
0000 39 MODIFIED BY:
0000 40
0000 41 V03-021 HWS0092 Harold Schultz 22-Jul-1984
0000 42 Add support for execute-only command procedures.
0000 43
0000 44 V03-020 HWS0077 Harold Schultz 29-Jun-9184
0000 45 If EOF encountered when reading a command procedure
0000 46 while processing a line continuation, ignore termination
0000 47 and return EOL.
0000 48
0000 49 V03-019 HWS0072 Harold Schultz 08-Jun-1984
0000 50 Partially undo HWS0006 (setting of the parse position).
0000 51 Set the parse position prior to processing the @FILESPEC
0000 52 construct, but save the old parse position set by the
0000 53 higher level routines and restore it after the 'a' processing
0000 54 has successfully completed.
0000 55
0000 56 V03-018 HWS0039 Harold Schultz 26-Mar-1984
0000 57 When using arrow keys, don't duplicate command when changing

```



0000 58 : directions in recalling commands and are at the beginning of  
0000 59 : the buffer.  
0000 60 :  
0000 61 : V03-017 HWS0034 Harold Schultz 16-Mar-1984  
0000 62 : Always use terminal RAB when outputting escape sequences  
0000 63 : and arrow keys.  
0000 64 :  
0000 65 : V03-016 HWS0006 Harold Schultz 13-Feb-1984  
0000 66 : Remove parse location setting (DCL\$MARK) when  
0000 67 : processing an indirect command file. Let higher  
0000 68 : level routines set parse location.  
0000 69 : Use PRC\_V\_CARRCNTL to determine whether or not  
0000 70 : a CR/LF is to be inserted before the prompt string.  
0000 71 : Fix RECALL\_NEXT so it will not skip first command  
0000 72 : at bottom of command buffer when recall buffer  
0000 73 : not full.  
0000 74 : Save RMS STS and STV values in PRC data table  
0000 75 :  
0000 76 : V03-015 PCG0017 Peter George 03-Jan-1984  
0000 77 : Move location of test for PRC\_V\_FLUSH.  
0000 78 :  
0000 79 : V03-014 PCG0017 Peter George 03-Jan-1984  
0000 80 : Use PRC\_V\_FLUSH to determine how to handle EOF when  
0000 81 : performing a flush.  
0000 82 :  
0000 83 : V03-013 PCG0016 Peter George 18-Nov-1983  
0000 84 : Support up and down arrow recall.  
0000 85 : Do not automatically close command procedures when  
0000 86 : flushing a record. Add support for erase keypad attribute.  
0000 87 :  
0000 88 : V03-012 PCG0015 Peter George 27-Sep-1983  
0000 89 : Ignore spurious CTRL/Y's.  
0000 90 : Only check for CTRL/B's and ESC if input is from terminal.  
0000 91 :  
0000 92 : V03-011 PCG0014 Peter George 17-Jul-1983  
0000 93 : Support 20 recalled commands.  
0000 94 :  
0000 95 : V03-010 PCG0013 Peter George 01-May-1983  
0000 96 : Correctly signal keypad buffer overflows.  
0000 97 : Fix CTRL/Y interrupting GOTO bug.  
0000 98 :  
0000 99 : V03-009 PCG0012 Peter George 20-Apr-1983  
0000 100 : Check for CTRL/B with CMPB.  
0000 101 :  
0000 102 : V03-008 PCG0011 Peter George 06-Apr-1983  
0000 103 : Remove GOTO code.  
0000 104 : Reformat DCL\$INPUT code.  
0000 105 : Change test for escape sequence.  
0000 106 : Allow the RECALL command to accept letters as arguments.  
0000 107 : Update recall buffer when at ctrl/y level.  
0000 108 :  
0000 109 : V03-007 PCG0010 Peter George 01-Apr-1983  
0000 110 : Change STV MOVZWL to MOVZBL.  
0000 111 :  
0000 112 : V03-006 PCG0009 Peter George 24-Feb-1983  
0000 113 : Do not verify lines in EXE-only command procedures.  
0000 114 : Lookup terminating escape sequences in the keypad

|      |     |      |  |
|------|-----|------|--|
| 0000 | 115 | :    | symbol table and act on them.                                |
| 0000 | 116 | :    | Use new XABTRM and init appropriate fields in its item list. |
| 0000 | 117 | :    | Add RECALL command and CTRL/B processing.                    |
| 0000 | 118 | :    |  |
| 0000 | 119 | :    | V03-005 PCG0008 Peter George 10-Feb-1983                     |
| 0000 | 120 | :    | Close SYS\$OUTPUT on silent logout.                          |
| 0000 | 121 | :    |  |
| 0000 | 122 | :    | V03-004 PCG0007 Peter George 15-Jan-1983                     |
| 0000 | 123 | :    | Supply more rigorous test of whether commands                |
| 0000 | 124 | :    | should be verified. Close PPF files before silent logout.    |
| 0000 | 125 | :    |  |
| 0000 | 126 | :    | V03-003 PCG0006 Peter George 28-Dec-1982                     |
| 0000 | 127 | :    | If \$GET fails because of insufficient quota,                |
| 0000 | 128 | :    | then log the process out.                                    |
| 0000 | 129 | :    |  |
| 0000 | 130 | :    | V03-002 PCG0005 Peter George 14-Nov-1982                     |
| 0000 | 131 | :    | Call DCL\$UPCASE instead of DCL\$REMBLANKS                   |
| 0000 | 132 | :    | Use prompt descriptor instead of WRK_L_PROMPT.               |
| 0000 | 133 | :    |  |
| 0000 | 134 | :    | V03-001 PCG0004 Peter George 28-Oct-1982                     |
| 0000 | 135 | :    | Get prompt string from PRC.                                  |
| 0000 | 136 | :    | Process escape sequences.                                    |
| 0000 | 137 | :--- |  |

```

0000 139 :
0000 140 : MACRO LIBRARY CALLS
0000 141 :
0000 142 :
0000 143 PRCDEF ;DEFINE PROCESS WORK AREA
0000 144 WRKDEF ;DEFINE COMMAND WORK AREA
0000 145 ITRMDEF ;XABTRM ITEM LIST DEFINITIONS
0000 146 SYMDEF ;DEFINE SYMBOL TABLE ENTRY FORMAT
0000 147 PTRDEF ;DEFINE TOKEN DESCRIPTORS
0000 148 $CLMSGDEF ;DEFINE ERROR/STATUS VALUES
0000 149 $FABDEF ;DEFINE FAB OFFSETS
0000 150 $RABDEF ;DEFINE RAB OFFSETS
0000 151 $XABTRMDEF ;DEFINE TERMINAL XAB
0000 152 $DVIDEF ;DEFINE $GETDVI ITEM CODES
0000 153 $TT2DEF ;DEFINE DEVDEPEND2 BITS
0000 154
00000000 155 .PSECT DCL$ZCODE, BYTE, RD, NOWRT
0000 156
04 0000 157 ERASE: .BYTE 4
4B5B1B0D 0001 158 .LONG ^X4B5B1B0D ;<CR>ESCCK

```



```
0005 160 .SBTTL READ NEXT INPUT RECORD
0005 161 ;+
0005 162 ; DCL$INPUT - READ NEXT INPUT RECORD
0005 163 ;
0005 164 ; READS THE NEXT INPUT RECORD AND PLACES IT INTO THE INPUT BUFFER.
0005 165 ; THE CHARACTER POINTER IS RESET TO THE BEGINNING OF THE RECORD. A
0005 166 ; SYMBOL SUBSTITUTION PASS IS PERFORMED IF A SINGLE OCCURRENCE OF A
0005 167 ; SINGLE QUOTE IS DETECTED IN THE RECORD.
0005 168 ;
0005 169 ; INPUTS:
0005 170 ;
0005 171 ; R11 = ADDRESS OF PRC AREA
0005 172 ; R10 = ADDRESS OF WRK AREA
0005 173 ;
0005 174 ; OUTPUTS:
0005 175 ;
0005 176 ; WRK_L_CHARPTR POINTS TO BEGINNING OF INPUT RECORD, WHICH
0005 177 ; HAS BEEN TERMINATED BY A NULL BYTE.
0005 178 ;
0005 179 ; R0 = FIRST CHARACTER IN INPUT BUFFER
0005 180 ;
0005 181 DCL$INPUT:: ; INPUT NEXT RECORD
1C BB 0005 182 PUSHF #^M<R2,R3,R4> ; SAVE REGISTERS
0007 183 ;
0007 184 ;
0007 185 ; IF AUTOLOGO FLAG SET AND WE ARE AT LEVEL 0 OR CTRL/Y LEVEL,
0007 186 ; THEN DELETE THIS PROCESS TO PERFORM AN IMPLIED LOGOUT BUT WITHOUT
0007 187 ; ANY LOGOUT MESSAGE. THIS IS FOR THE SPAWN COMMAND.
0007 188 ;
12 68 AB 08 E1 0007 189 REINP: BBC #PRC_V_AUTOLOGO,PRC_W_FLAGS(R11),20$ ;BRANCH IF FLAG NOT SET
0D F0 AA 07 E0 000C 190 BBS #WRK_V_INQUIRE,WRK_W_FLAGS(R10),20$ ;BRANCH IF INQUIRE
05 68 AB 0B E0 0011 191 BBS #PRC_V_YLEVEL,PRC_W_FLAGS(R11),10$ ;IF SET, AT CONTROL Y/C LEVE
5C AB D5 0016 192 TSTL PRC_C_INDEPTH(R11) ;INDIRECT LEVEL ZERO?
03 12 0019 193 BNEQ 20$ ;BRANCH IF NOT
027D 31 001B 194 10$: BRW SILENT_LOGOUT ;PERFORM SILENT LOGOUT
001E 195 ;
001E 196 ;
001E 197 ; GET ADDRESS OF THE RAB ASSOCIATED WITH THIS INDIRECT LEVEL
001E 198 ;
0E F0 AA 07 E0 001E 199 20$: BBS #WRK_V_INQUIRE,WRK_W_FLAGS(R10),30$ ;IF SET, QUERY IN PROGRESS
54 14 AB D0 0023 200 MOVL PRC_C_INDINPRAB(R11),R4 ;GET ADDRESS OF LEVEL N RAB
09 68 AB 0B E1 0027 201 BBC #PRC_V_YLEVEL,PRC_W_FLAGS(R11),40$ ;IF CLR, NOT AT CONTROL Y/C
04 68 AB 04 E0 002C 202 BBS #PRC_V_GOTO,PRC_W_FLAGS(R11),40$ ;IF SET, AT GOTO LEVEL
54 08 AB D0 0031 203 30$: MOVL PRC_C_INPRAB(R11),R4 ;GET ADDRESS OF LEVEL 0 RAB
0035 204 ;
0035 205 ;
0035 206 ; SETUP PROMPT STRING
0035 207 ;
0C 50 10 AB D0 0035 208 40$: MOVL PRC_L_TRMLIST(R11),R0 ;GET ADDRESS OF XABTRM ITEM
0C A0 F99E CA B0 0039 209 MOVW WRK_W_PMPTLEN(R10),ITRM_W_PMPTLEN(R0) ;SET LENGTH OF PROMPT STRING
10 A0 F9A2 CA D0 003F 210 MOVL WRK_L_PMPTADDR(R10),ITRM_C_PMPTADDR(R0) ;SET ADDRESS OF PROMPT STRIN
0045 211
```



```
0045 213 :  
0045 214 : SETUP INPUT BUFFER AND POINTERS.  
0045 215 :  
0045 216 GET_INPUT:  
20 A4 0100 8F B0 0045 217 MOVW #WRK_C_INPBUFSIZ,RAB$W_USZ(R4) :SET SIZE OF INPUT BUFFER  
52 F894 CA 9E 004B 218 MOVAB WRK_G_INPBUF-2(R10),R2 :GET ADDRESS OF INPUT BUFFER  
24 A4 52 D0 0050 219 MOVL R2,RAB$L_UBF(R4) :SET ADDRESS OF INPUT BUFFER  
F48E CA FF A2 9E 0054 220 MOVAB -1(R2),WRK_L_CHARPTR(R10) :SET POINTER FOR GET CHARACT  
005A 221 :  
005A 222 :  
005A 223 : READ THE NEXT INPUT RECORD AND CHECK FOR ERRORS.  
005A 224 :  
05 F0 AA 07 E1 005A 225 :  
005E 226 DISABLE :DISABLE CONTROL Y/C AST'S  
0063 227 BBC #WRK_V_INQUIPE,WRK_W_FLAGS(R10),10$ :SKIP IF NOT INQUIRING  
0068 228 10$: $GET RAB$V_PPF_IND,RAB$W_ISI(R4) :SET INDIRECT PPF  
0071 229 CLRBIT RAB$V_PPF_IND,RAB$W_ISI(R4) :GET NEXT RECORD FROM INPUT  
23 68 AB 01 E1 0076 230 BBC #PRC_V_CNTRLY,PRC_W_FLAGS(R11),20$ :CLEAR INDIRECT PPF  
1A 18 A4 00000000'8F E1 007B 231 BBC #DEV$V_TRM,RAB$L_CTR(R4),20$ :BRANCH IF NO CTRL/Y PENDING  
00000000'8F 50 D1 0084 232 CMPL R0,#RMS$_CONTROLY :SKIP IF NOT TERMINAL  
OD 12 008B 233 BNEQ 15$ :DOUBLE CHECK FOR WINDOW  
51 40 A4 D0 008D 234 MOVL RAB$L_XAB(R4),R1 :SPURIOUS CTRL/Y  
18 DO 0091 235 MOVL #ITRM_K_MINLEN,- :GET ADDRESS OF XABTRM  
OC A1 0093 236 XAB$W_ITMLST_LEN(R1) :SET SHORT LENGTH OF ITEM LI  
0519 30 0095 237 BSBW DCL$LOCKED_STATE :RESTORE LOCKED STATE  
04 11 0098 238 BRB 20$ :  
009A 239 15$: CLRBIT PRC_V_CNTRLY,PRC_W_FLAGS(R11) :IGNORE THIS CTRL/Y  
009E 240 20$: ENABLE :ENABLE CONTROL Y/C AST'S  
OE 50 E8 00A0 241 BLBS R0,PROCESS_INPUT :IF LBC I/O ERROR  
0000'8F 50 B1 00A3 242 CMPW R0,#RMS$_SYS&^XFFFF :ERROR IN QIO?  
04 12 00AB 243 BNEQ ERROR1 :NO, THEN SKIP  
50 OC A4 D0 00AA 244 MOVL RAB$L_STV(R4),R0 :GET PARTICULAR ERROR STATUS  
00EA 31 00AE 245 ERROR1: BRW IO_ERROR :ERROR  
00B1 246
```

```
00B1 248 :  
00B1 249 : READ COMPLETED SUCCESSFULLY. CLEAN UP AND THEN PROCESS THE INPUT RECORD.  
00B1 250 :  
00B1 251 PROCESS_INPUT:  
53 22 A4 3C 00B1 252 MOVZWL RAB$W_RSZ(R4),R3 ;GET LENGTH OF INPUT RECORD  
51 40 A4 DO 00B5 253 MOVL RAB$L_XAB(R4),R1 ;GET ADDRESS OF XABTRM  
18 DO 00B9 254 MOVL #ITRM_K_MINLEN,- ;SET SHORT LENGTH OF ITEM LI  
0C A1 00BB 255 XAB$W_ITMLST_LEN(R1) ;  
50 10 AB DO 00BD 256 MOVL PRC_L_TRMLIST(R11),R0 ;GET ADDRESS OF XABTRM ITEM  
06 68 AB 00 E1 00C1 257 BBC #PRC_V_CARRCNTL,PRC_W_FLAGS(R11),2$ ;SKIP IF NO CR/LF INDICATED  
10 B0 0000'CF B0 00C6 258 MOVW DCL$CRF,@ITRM_L_PMPADDR(R0) ;INSERT A CR/LF OTHERWISE  
00CC 259 :  
00CC 260 : IF IN THE MIDST OF A GOTO SCAN, THEN SKIP THE UNNECESSARY PROCESSING.  
00CC 261 :  
09 68 AB 04 E1 00CC 262 2$: BBC #PRC_V_GOTO,PRC_W_FLAGS(R11),10$ ;SKIP IF NOT GOTO READ  
6243 94 00D1 263 CLR (R2)[R3] ;SET EOL  
00A8 31 00D4 264 BRW RETURN ;RETURN  
FF6B 31 00D7 265 5$: BRW GET_INPUT ;  
00DA 266 :  
00DA 267 : CHECK FOR CTRL/B AND DEFINED KEYS.  
00DA 268 :  
16 18 A4 00000000'8F E1 00DA 269 10$: BBC #DEV$V_TRM,RAB$L_CTX(R4),12$ ;SKIP IF NOT TERMINAL  
01DF 30 00E3 271 BSBW PROCESS_RECALL ;CHECK FOR RECALL CHAR  
03 50 D1 00E4 272 CMPL R0,#3 ;REPROMPT REQUESTED?  
EC 13 00E9 273 BEQL 5$ ;YES, DO IT  
0319 30 00EB 274 BSBW PROCESS_ESCAPE ;PROCESS ESCAPE SEQUENCES  
03 50 D1 00EE 275 CMPL ^0,#3 ;REPROMPT REQUESTED?  
E4 13 00F1 276 BEQL 5$ ;YES, DO IT  
6243 94 00F3 277 CLR (R2)[R3] ;SET EOL INDICATOR  
B5 50 E9 00F6 278 BLBC R0,ERROR1 ;BRANCH IF ERROR  
00F9 279 :  
00F9 280 :  
00F9 281 : IF INTERACTIVE AND IF THE RECORD IS NON-NULL, THEN COPY THE COMMAND LINE  
00F9 282 : TO THE RECALL BUFFER.  
00F9 283 :  
6243 94 00F9 284 12$: CLRB (R2)[R3] ;SET EOL INDICATOR  
53 D5 00FC 285 TSTL R3 ;NULL COMMAND?  
31 13 00FE 286 BEQL 40$ ;YES, THEN SKIP  
2C 68 AB 06 E0 0100 287 BBS #PRC_V_MODE,PRC_W_FLAGS(R11),40$ ;IF SET, BATCH JOB  
0A 68 AB 0B E0 0105 288 BBS #PRC_V_YLEVEL,PRC_W_FLAGS(R11),15$ ;IF SET, CTRL/Y LEVEL  
05 F0 AA 07 E0 010A 289 BBS #WRK_V_INQUIRE,WRK_W_FLAGS(R10),15$ ;IF SET, INQUIRING  
5C AB D5 010F 290 TSTL PRC_L_INDEPTH(R11) ;INTERACTIVE LEVEL 0?  
1D 12 0112 291 BNEQ 40$ ;IF NO, DON'T SAVE  
52 DD 0114 292 15$: PUSHL R2 ;PUSH COMMAND ADDR  
53 DD 0116 293 PUSHL R3 ;PUSH COMMAND LEN  
5E DD 0118 294 PUSHL SP ;PUSH DESCR ADDR  
07 F0 AA 03 E0 011A 295 BBS #WRK_V_CONTIN,WRK_W_FLAGS(R10),20$ ;IS IT CONTIN TYPE PROMPT?  
0000'CF 01 FB 011F 296 CALLS #1,DCL$PUT_COMMAND ;SAVE THE COMMAND AWAY  
05 11 0124 297 BRB 30$ ;BRANCH  
0000'CF 01 FB 0126 298 20$: CALLS #1,DCL$PUT_SEGMENT ;SAVE THE SEGMENT AWAY  
53 8ED0 012B 299 30$: POPL R3 ;RESTORE R3  
52 8ED0 012E 300 POPL R2 ;RESTORE R2  
0131 301 :  
0131 302 :  
0131 303 : SUBSTITUTE ANY SYMBOLS WHICH ARE DELIMITED BY SINGLE QUOTES  
0131 304 :
```



```
62 53 27 3A 0131 305 40$: LOCC #^A/'/,R3,(R2) ;LINE HAVE POSSIBLE STRING S
      03 13 0135 306 BEQL 50$ ;IF EQL NO
      0490 30 0137 307 BSBW EXPAND ;EXPAND LINE IF APPROPRIATE
      013A 308
      013A 309
      013A 310 : IF VERIFY MODE, WRITE A COPY OF THE COMMAND LINE TO THE LOG FILE
      013A 311
      1B 68 AB 07 E1 013A 312 50$: BBC #PRC_V_VERIFY,PRC_W_FLAGS(R11),70$ ;IF CLR, NO LINE VERIFICATION
      16 68 AB 0B E0 013F 313 BBS #PRC_V_YLEVEL,PRC_W_FLAGS(R11),70$ ;IF SET, AT CONTROL Y/C LEVE
      0B 68 AB 06 E0 0144 314 BBS #PRC_V_MODE,PRC_W_FLAGS(R11),60$ ;IF SET, BATCH JOB
      SC AB D5 0149 315 TSTL PRC_C_INDEPTH(RT1) ;INTERACTIVE LEVEL 0?
      OC 13 014C 316 BEQL 70$ ;IF YES, DON'T ECHO
      012D CB 95 014E 317 TSTB PRC_B_EXONLYL(R11) ;EXE-ONLY PROCEDURE?
      06 12 0152 318 BNEQ 70$ ;IF YES, DON'T ECHO
      51 53 D0 0154 319 60$: MOVL R3,R1 ;THE LENGTH OF THE LINE
      FEA6' 30 0157 320 BSBW DCL$MSGOUT ;OUTPUT INPUT LINE
      015A 321
      015A 322
      015A 323 : IF WE JUST READ A FULL-LINE COMMENT, RE-ISSUE READ NOW AS AN OPTIMIZATION.
      015A 324
      20 68 AB 01 E0 015A 325 70$: BBS #PRC_V_CNTRLY,PRC_W_FLAGS(R11),RETURN ;MUST TAKE IMMEDIATE ACTION
      54 D4 015F 326 CLRL R4 ;IF CTRL HIT - SKIP OPTIMIZ
      96'AF 05 82 3A 0161 328 80$: LOCC (R2)+,#5,B^SPECIAL ;SET NO DOLLAR SIGN SEEN FLA
      17 13 0166 329 BEQL RETURN ;CHECK FOR SPECIAL CHARS
      50 03 C2 0168 330 SUBL #3,R0 ;IF EQL NO MATCH
      F4 19 016B 331 BLSS 80$ ;BLANK OR TAB?
      10 13 016D 332 BEQL RETURN ;IF LSS YES
      50 D7 016F 333 DECL R0 ;IF EQL END OF LINE
      08 1B 0171 334 BLEQU 90$ ;DOLLAR SIGN OR COMMENT?
      07 68 AB 05 E0 0173 335 BBS #PRC_V_IND,PRC_W_FLAGS(R11),RETURN ;IF LEQU DOLLAR SIGN
      FE8C 31 0178 336 BRW REINP ;BRANCH IF FLUSHING RECORD
      E2 54 00 E3 017B 337 90$: BBS #0,R4,80$ ;GET NEXT RECORD
      017F 338 ;IF CLR, FIRST DOLLAR SIGN
      017F 339
      017F 340 : IF THE PREVIOUS RECORD ENDED WITH TRAILING SPACES OR TABS,
      017F 341 : INSERT A SPACE AT THE FRONT OF THE CURRENT INPUT RECORD SO
      017F 342 : THAT PARAMETERS ARE DELIMITED PROPERLY.
      017F 343
      06 F0 AA 09 E5 017F 344 RETURN: BBCC #WRK_V_TRAILSPC,WRK_W_FLAGS(R10),10$ ;IF CLR, NO TRAILING SPACE S
      50 20 90 0184 345 MOVB #^A/'/,R0 ;SET SPACE CHARACTER
      FE76' 30 0187 346 BSBW DCL$BACKUPCHAR ;APPEND TO FRONT OF INPUT BU
      1C BA 018A 347 10$: POPR #^M<R2,R3,R4> ;RESTORE REGISTERS
      00F3 CB 5F 8F 90 018C 348 MOVB #^A/'/,PRC_B_CONTINUE(R11) ;SET FOR CONTINUATION PROMPT
      FE6B' 30 0192 349 BSBW DCL$GETCHAR ;GET FIRST CHARACTER OF NEW
      05 0195 350 RSB
      0196 351
      0196 352 SPECIAL:
      09 20 00 24 21 0196 353 .ASCII /!$/<0>/ / ;SPECIAL CHARACTERS
```



```
019B 355 : AN INPUT I/O ERROR HAS OCCURRED. IF WE GOT 'RECORD STREAM ACTIVE',
019B 356 : TRY REPEATING THE READ 1000 TIMES. IF THAT FAILS, TRY CANCELING ALL
019B 357 : I/O ON THE INPUT CHANNEL AND THEN REISSUING THE READ.
019B 358 :
019B 359 :
019B 360 : I/O_ERROR:
019B 361 : CMPW R0,#RMS$_RSAB^XFFFF ;ERROR RECORD STREAM ACTIVE
01A0 362 : BNEQ 30$ ;IF NO CHECK FOR END_OF_FILE
01A2 363 :
01A2 364 :
01A2 365 : WE DON'T WANT TO HAVE TO CANCEL AN RMS I/O UNLESS NECESSARY.
01A2 366 : RETRY UP SEVERAL TIMES TO ALLOW CURRENT WRITE I/O TO COMPLETE.
01A2 367 : THIS AVOIDS SPURIOUS WRITE ABORT MESSAGES FROM USER PROGRAMS,
01A2 368 : CAUSED BY CANCELING AN RMS I/O OPERATION.
01A2 369 :
01A2 370 : MOVZWL #1000,R2 ;SETUP RETRY COUNT
01A7 371 : BBC #WRK_V_INQUIRE,WRK_W_FLAGS(R10),10$ ;SKIP IF NOT INQUIRING
01AC 372 : SETBIT RAB$V_PPF_IND,RAB$W_ISI(R4) ;SET INDIRECT PPF
01B1 373 : GET RAB=(R4) ;SEE IF THE WAITING GAVE RMS
01BA 374 : CLRBIT RAB$V_PPF_IND,RAB$W_ISI(R4) ;CLEAR INDIRECT PPF
01BF 375 : BLBS R0,20$ ;THE WAIT DID THE TRICK, GO
01C2 376 : CMPW R0,#RMS$_RSAB^XFFFF ;RECORD STREAM STILL ACTIVE?
01C7 377 : BNEQ 30$ ;BRANCH IF NOT
01C9 378 : SOBGTR R2,10$ ;RETRY UNTIL STREAM BECOMES
01CC 379 :
01CC 380 :
01CC 381 : CONSTANT RETRYING DIDN'T HELP. CANCEL ANY I/O AND TRY AGAIN.
01CC 382 :
01CC 383 : $CANCEL_S_PRC_W_INPCHAN(R11) ;IF NO LUCK, STOP THE I/O ON
01D7 384 : $WAIT RAB=(R4) ;WAIT FOR I/O TO COMPLETE
01E0 385 : BRB 40$ ;TRY TO READ AGAIN
01E2 386 :
01E2 387 :
01E2 388 : READ FINALLY SUCCEEDED. PROCESS THE INPUT RECORD.
01E2 389 :
01E2 390 : 20$: MOVAB WRK_G_INPBUF-2(R10),R2 ;GET ADDRESS OF INPUT BUFFER
01E7 391 : BRW PROCESS_INPUT ;PROCESS THE INPUT LINE
01EA 392 :
01EA 393 :
01EA 394 : WE HAVE ENCOUNTERED AN ERROR OTHER THAN RECORD STREAM ACTIVE. RESTORE THE
01EA 395 : DEFAULT READ FORMAT OF NO INITIAL STRING, NO OFFSET AND RESTORE THE DEFAULT
01EA 396 : KEYPAD STATE.
01EA 397 :
01EA 398 : 30$: MOVL RAB$L_XAB(R4),R1 ;GET ADDRESS OF XABTRM
01EE 399 : MOVL #ITRM_K_MINLEN,XAB$W_ITMLST_LEN(R1) ;SET SHORT LENGTH OF ITEM LI
01F2 400 : BSBW DCL$LOCKED_STATE ;RESTORE LOCKED KEYPAD STATE
01F5 401 :
01F5 402 :
01F5 403 : IF WE ARE AT END OF FILE, TERMINATE THE CURRENT PROCEDURE LEVEL AND
01F5 404 : READ THE NEXT RECORD FROM THE PREVIOUS PROCEDURE LEVEL.
01F5 405 :
01F5 406 : CMPW R0,#RMS$_EOF^XFFFF ;END OF FILE?
01FA 407 : BNEQ 110$ ;IF NEQ NO
01FC 408 : CLRB (R2) ;SET END OF LINE INDICATOR
01FE 409 : BBS #PRC_V_YLEVEL,PRC_W_FLAGS(R11),50$ ;IF SET, AT CONTROL Y/C LEVE
0203 410 : BBS #WRK_V_INQUIRE,WRK_W_FLAGS(R10),50$ ;BR IF DOING AN INQUIRE
0208 411 : TSTL PRC_C_INDEPTH(R11) ;INDIRECT LEVEL ZERO?
```

0000'8F 50 B1 019B 355  
48 12 01A0 356  
01A2 357  
01A2 358  
01A2 359  
01A2 360  
01A2 361  
01A2 362  
01A2 363  
01A2 364  
01A2 365  
01A2 366  
01A2 367  
01A2 368  
01A2 369  
01A2 370  
05 F0 AA 07 E1 01A7 371  
01AC 372  
01B1 373  
01BA 374  
20 50 E8 01BF 375  
0000'8F 50 B1 01C2 376  
21 12 01C7 377  
E5 52 F5 01C9 378  
01CC 379  
01CC 380  
01CC 381  
01CC 382  
01CC 383  
37 11 01D7 384  
01E0 385  
01E2 386  
01E2 387  
01E2 388  
01E2 389  
52 F894 CA 9E 01E2 390  
FEC7 31 01E7 391  
01EA 392  
01EA 393  
01EA 394  
01EA 395  
01EA 396  
01EA 397  
51 40 A4 D0 01EA 398  
OC A1 18 D0 01EE 399  
03BC 30 01F2 400  
01F5 401  
01F5 402  
01F5 403  
01F5 404  
01F5 405  
0000'8F 50 B1 01F5 406  
58 12 01FA 407  
62 94 01FC 408  
28 68 AB 0B E0 01FE 409  
23 F0 AA 07 E0 0203 410  
5C AB D5 0208 411

```
21 13 020B 412 BEQL 70$ ;IF EOL, YES
FDFO' 30 020D 413 BSBW DCL$UNSTACK ;UNSTACK INDIRECT FILE
0210 414
0210 415
0210 416 : IF WE JUST RETURNED BACK TO LEVEL 0, FORCE THE PROMPT STRING BACK TO NORMAL
0210 417 : BEFORE RE-ISSUING A READ FOR THE NEXT COMMAND.
0210 418
5C AB 05 0210 419 TSTL PRC_L_INDEPTH(R11) ;INDIRECT LEVEL ZERO?
04 12 0213 420 BNEQ 40$ ;BRANCH IF NOT
00F3 CB 94 0215 421 CLR B PRC_B_CONTINUE(R11) ;SET FOR NORMAL PROMPT
0219 422
0219 423
0219 424 : IF CURRENTLY PROCESSING A LINE CONTINUATION, RETURN AN EOL CHARACTER.
0219 425
0219 426 : IF INDIRECT FILE RECOGNITION IS DISABLED, THIS IS A FLUSH OF A COMMAND WITH
0219 427 : "-" AS THE LAST CHARACTER. RETURN AN EOL CHARACTER. OTHERWISE, PROCESS
0219 428 : THE LINE JUST READ
0219 429
0D 68 AB 06 E0 0219 430 40$: BBS #PRC_V_FLUSH,PRC_W_FLAGS(R11),50$ ;IF BIT IS SET,THEN FLUSH
08 68 AB 05 E0 021E 431 BBS #PRC_V_IND,PRC_W_FLAGS(R11),50$ ;IF BIT IS SET,THEN FLUSH
03 FO AA 03 E0 0223 432 BBS #WRK_V_CONTIN,WRK_W_FLAGS(R10),50$ ;IF BIT IS SET,THEN FLUSH
FDDC 31 0228 433 BRW REINP ;READ THE NEXT LINE
FF51 31 022B 434 50$: BRW RETURN ;FLUSH THE RECORD
022E 435
022E 436
022E 437 : IF WE GOT AN END OF FILE WHILE READING THE LEVEL 0 PROCEDURE IN A
022E 438 : NON-INTERACTIVE JOB, TERMINATE THE JOB STEP. IGNORE EOF'S (CTRL/Z)
022E 439 : IN AN INTERACTIVE JOB.
022E 440
14 68 AB 04 E5 022E 441 70$: BBCC #PRC_V_GOTO,PRC_W_FLAGS(R11),90$ ;BR IF NOT IN A GOTO
FDCA' 30 0233 442 BSBW DCL$DEALGOTO ;DEALLOCATE GOTO SYMBOL
0236 443 STATUS USGOTO ;SET FINAL STATUS OF UNSTATI
023D 444 SETBIT WRK_V_COMMAND,WRK_W_FLAGS(R10) ;MARK COMMAND EXECUTION ERRO
0241 445 ERRMSG ;PRINT THE ERROR
68 AB FDB9' 30 0244 446 80$: BSBW DCL$SET STATUS ;GIVE ERROR HANDLER A CHANCE
0E 0E 0247 447 90$: BBS #PRC_V_EOFLOGO,PRC_W_FLAGS(R11),- ;IF SILENT LOGOUT REQUESTED
4F 024B 448 ;SILENT LOGOUT
CB 68 AB 06 E1 024C 449 BBC #PRC_V_MODE,PRC_W_FLAGS(R11),40$ ;IF NOT BATCH JOB, IGNORE EO
FDAC' 31 0251 450 BRW DCL$ABORT ;LOG OUT BATCH JOB
0254 451
0254 452
0254 453 : SOME OTHER TYPE OF I/O ERROR HAS OCCURRED. ISSUE AN ERROR MESSAGE,
0254 454 : THEN TERMINATE THE CURRENT PROCEDURE LEVEL AND PARSE THE NEXT COMMAND.
0254 455 : IF THE ERROR WAS INSUFFICIENT QUOTA, THEN CANCEL THE CTRL/Y AST, OUTPUT
0254 456 : THE ERROR MESSAGE, AND THEN LOG THE PROCESS OUT.
0254 457
00000000'8F 50 D1 0254 458 110$: SETBIT WRK_V_COMMAND,WRK_W_FLAGS(R10) ;MARK COMMAND EXECUTION ERRO
2C 13 0258 459 CMPL RO,755$_EXQUOTA ;IS ERROR DUE TO EXCEEDED QU
025F 460 BEQL ABORT ;YES, THEN BRANCH
0261 461 ASSUME PRC_L_STV EQ PRC_L_STS+4
0261 462 ASSUME RAB$L_STV EQ RAB$L_STS+4
0084 CB 08 A4 7D 0261 463 MOVQ RAB$L_STS(R4),PRC_L_STS(R11) ;STORE STS AND STV VALUES
0267 464 120$: ERRMSG ;OUTPUT ERROR MESSAGE
18 68 AB 08 E0 026A 465 BBS #PRC_V_YLEVEL,PRC_W_FLAGS(R11),STATUS ;IF SET, CNTL Y/C LEVEL
13 FO AA 07 E0 026F 466 BBS #WRK_V_INQUIRE,WRK_W_FLAGS(R10),STATUS ;SKIP IF IN INQUIRE
SC AB 05 0274 467 TSTL PRC_L_INDEPTH(R11) ;INDIRECT LEVEL ZERO?
07 07 0277 468 BNEQ 130$ ;IF NEQ NO
```

READREC  
V04-000

- READ AN INPUT RECORD  
READ NEXT INPUT RECORD

F 8

16-SEP-1984 00:11:48 VAX/VMS Macro V04-00  
4-SEP-1984 23:42:34 [DCL.SRC]READREC.MAR;1

Page 11  
(6)

|          |    |      |      |         |       |                                   |                             |
|----------|----|------|------|---------|-------|-----------------------------------|-----------------------------|
| C6 6B AB | 06 | E0   | 0279 | 469     | BBS   | #PRC V_MODE,PRC_W_FLAGS(R11),80\$ | :BR IF BATCH                |
|          | 07 | 11   | 027E | 470     | BRB   | STATUS                            | :                           |
|          | 50 | DD   | 0280 | 471     | PUSHL | RO                                | :SAVE ERROR/STATUS VALUE    |
| FD7B'    | 30 | 0282 | 472  | 130\$:  | BSBW  | DCL\$UNSTACK                      | :UNSTACK INDIRECT FILE      |
| 01       | BA | 0285 | 473  |         | POPR  | #*M<RO>                           | :RESTORE ERROR/STATUS VALUE |
| FD76'    | 30 | 0287 | 474  | STATUS: | BSBW  | DCL\$SET STATUS                   | :SET COMPLETION STATUS      |
| FD73'    | 31 | 028A | 475  |         | BRW   | DCL\$RESTART                      | :                           |
|          |    | 028D | 476  |         |       |                                   | :                           |



```

028D 478 ::
028D 479 :: OUTPUT AN ERROR MESSAGE AND LOG THE PROCESS OUT
028D 480 ::
50 DD 028D 481 ABORT: PUSHL R0 ;SAVE THE STATUS
FD6E' 30 028F 482 BSBW DCL$DSBCONTRLY ;CANCEL THE CTRL/Y AST
50 BED0 0292 483 POPL R0 ;RESTART THE STATUS
FD65' 31 0295 484 ERRMSG ;OUTPUT THE ERROR MESSAGE
0298 485 BRW DCL$ABORT ;LOG THE PROCESS OUT
0298 486 ::
0298 487 ::
0298 488 :: PERFORM A SILENT LOGOUT BY CANCELING THE SUPERVISOR MODE EXIT HANDLERS
0298 489 (SO THAT THE PROCESS IS DELETED), AND INVOKING $EXIT WITH THE LATEST
0298 490 STATUS FOR THIS PROCESS.
0298 491 ::
0298 492 SILENT_LOGOUT::
50 FD62' 30 0298 493 BSBW DCL$CLOSE PPFS ;CLOSE ALL PPF FILES STILL O
1C AB D0 029E 494 MOVL PRC_L_INDFAB(R11),R0 ;GET ADDR OF INDIRECT FAB
0114 CB B0 02A2 495 MOVW PRC_W_OUTIFI(R11),- ;GET INTERNAL FILE INDEX OF
02 A0 02A6 496 FAB$W-IFI(R0)
02A8 497 $CLOSE FAB=(R0) ;CLOSE INDIRECT OUTPUT FILE
02B1 498 $CANEXH_S ;CANCEL SUPERVISOR MODE EXIT
02BA 499 $EXIT_S-PRC_L_LSTSTATUS(R11) ;EXIT PROCESS WITH FINAL STA

```

```
02C5 501 .SBTTL PROCESS RECALL COMMANDS
02C5 502 ---
02C5 503 PROCESS_RECALL - PROCESS RECALL COMMANDS
02C5 504
02C5 505 SUBROUTINE TO CHECK FOR AND PROCESS CTRL/B AND ARROW RECALL COMMANDS.
02C5 506
02C5 507 INPUTS:
02C5 508
02C5 509 R2 = ADDRESS OF NEW INPUT RECORD
02C5 510 R3 = LENGTH OF NEW INPUT RECORD, EXCLUDING NULL AT END OF LINE
02C5 511 R4 = ADDRESS OF RAB
02C5 512
02C5 513 OUTPUTS:
02C5 514
02C5 515 INPUT BUFFER IS INITIALIZED
02C5 516 RO IS SET
02C5 517 ---
02C5 518 PROCESS_RECALL: ;PROCESS RECALL COMMAND
02C5 519
02C5 520
02C5 521 CHECK FOR CTRL/B.
02C5 522
02C5 523 CMPB #X02,RAB$W_STV0(R4) ;TERMINATOR CTRL/B?
02C5 524 BEQL RECALL_PREV ;YES, THEN PROCESS IT
02C5 525 MOVL #1,R0 ;RETURN
02C5 526 RSB
02C5 527
02C5 528
02C5 529 REPROMPT WITH THE PREVIOUS COMMAND.
02C5 530
02C5 531 RECALL_PREV:
02C5 532 BSBW DCL$LOCKED STATE ;RESTORE LOCKED STATE
02C5 533 TSTB WRK_B_RECALLCNT(R10) ;FIRST TIME?
02C5 534 BEQL 10$ ;YES, THEN ALWAYS GO AHEAD
02C5 535 TSTB @WRK_L_RECALLPTR(R10) ;POINTER ADJUST. BY RECALL_NEXT?
02C5 536 BEQL 5$ ;NO, OK AS IS
02C5 537 DECL WRK_L_RECALLPTR(R10) ;YES, SET IT BACK TO NORMAL
02C5 538
02C5 539 5$: CMPL PRC_L_RECALLPTR(R11),- ;RETURNED TO THE ORIGIN?
02C5 540 WRK_L_RECALLPTR(R10)
02C5 541 BEQL END_OF_LIST
02C5 542 CMPB WRK_B_RECALLCNT(R10),- ;MAX # OF COMMANDS DISPLAYED?
02C5 543 #WRK_C_RECALLMAX+1
02C5 544 BEQL END_OF_LIST
02C5 545 10$: CLRQ -(SP) ;ALLOCATE A DESCRIPTOR
02C5 546 PUSHL SP ;PUSH DESCR ADDRESS
02C5 547 CALLS #1,DCL$GET_PREV_COMMAND ;RECALL THE SPECIFIED COMMAND
02C5 548 BLBC R0,20$ ;BRANCH IF NO COMMAND WAS FOUND
02C5 549 INCB WRK_B_RECALLCNT(R10) ;INCR RECALL COUNT
02C5 550 MOVAB PRC_G_COMMANDS(R11),R0 ;R0 = ADDR. OF BEGINNING OF BUFFER
02C5 551 CMPL R0,WRK_L_RECALLPTR(R10) ;ARE WE AT BEGINNING OF BUFFER?
02C5 552 BNEQ 15$
02C5 553 INCL WRK_L_RECALLPTR(R10) ;YES, FIX POINTER TO INDICATE
02C5 554 15$: CMPB WRK_B_RECALLCNT(R10),- ;THIS TO RECALL NEXT.
02C5 555 #WRK_C_RECALLMAX+1 ;MAX # OF COMMANDS DISPLAYED?
02C5 556 BNEQ RECALL_CURR ;BRANCH IF NOT NOW MAX
02C5 557
```

OC A4 02 91 02C5 523  
04 13 02C9 524  
50 01 D0 02CB 525  
05 05 02CE 526  
02CF 527  
02CF 528  
02CF 529  
02CF 530  
02CF 531  
02DF 30 02CF 532  
C5 AA 95 02D2 533  
16 13 02D5 534  
EA BA 95 02D7 535  
03 13 02DA 536  
EA AA D7 02DC 537  
012F CB D1 02DF 538  
EA AA 13 02E3 539  
3B 13 02E5 540  
C5 AA 91 02E7 541  
15 13 02EA 542  
35 7C 02EB 543  
7E DD 02ED 544  
5E FB 02EF 545  
0000'CF 01 02F1 546  
17 50 E9 02F6 547  
C5 AA 96 02F9 548  
50 0133 CB 9E 02FC 549  
EA AA 50 D1 0301 550  
03 12 0305 551  
EA AA D6 0307 552  
030A 553  
C5 AA 91 030A 554  
15 030D 555  
05 12 030E 556

```
SE 08 C0 0310 558 20$: ADDL #8,SP ;RESTORE THE STACK
OD 11 0311 559 BRB END_OF_LIST ;OUTPUT A BLANK LINE
0312 560
0313 561
0314 562
0315 563
0316 564
0317 565
0318 566
0319 567
031A 568
031B 569
031C 570
031D 571
031E 572
031F 573
0320 574
0321 575
0322 576
0323 577
0324 578
0325 579
0326 580
0327 581
0328 582
0329 583
032A 584
032B 585
032C 586
032D 587
032E 588
032F 589
0330 590
0331 591
0332 592
0333 593
0334 594
0335 595
0336 596
0337 597
0338 598
0339 599
033A 600
033B 601
033C 602
033D 603
033E 604
033F 605
0340 606
0341 607
0342 608
0343 609
0344 610
0345 611
0346 612
0347 613
0348 614

0067 30 0315 565 RECALL_CURR:
51 8E 7D 0318 566 BSBW ERASE_LINE ;ERASE THE PREVIOUS LINE
00D1 30 031B 567 MOVQ (SP)+,R1 ;GET COMMAND DESCRIPTOR
50 03 D0 031E 568 BSBW INSERT_COMMAND ;MODIFY THE XABTRM
05 05 0321 569 MOVL #3,R0 ;SET REPROMPT RETURN STATUS
0322 570 RSB ;RETURN
0323 571
0324 572
0325 573
0326 574
0327 575
0328 576
0329 577
032A 578
032B 579
032C 580
032D 581
032E 582
032F 583
0330 584
0331 585
0332 586
0333 587
0334 588
0335 589
0336 590
0337 591
0338 592
0339 593
033A 594
033B 595
033C 596
033D 597
033E 598
033F 599
0340 600
0341 601
0342 602
0343 603
0344 604
0345 605
0346 606
0347 607
0348 608
0349 609
034A 610
034B 611
034C 612
034D 613
034E 614

005A 30 0322 575 END_OF_LIST:
51 7C 0323 576 BSBW ERASE_LINE ;ERASE THE PREVIOUS LINE
00C5 30 0327 577 CLRQ R1 ;INIT COMMAND DESCRIPTOR
50 03 D0 032A 578 BSBW INSERT_COMMAND ;REPROMPT WITH THIS COMMAND
05 05 032D 579 MOVL #3,R0 ;SET REPROMPT RETURN STATUS
032E 580 RSB ;RETURN
032F 581
0330 582
0331 583
0332 584
0333 585
0334 586
0335 587
0336 588
0337 589
0338 590
0339 591
033A 592
033B 593
033C 594
033D 595
033E 596
033F 597
0340 598
0341 599
0342 600
0343 601
0344 602
0345 603
0346 604
0347 605
0348 606
0349 607
034A 608
034B 609
034C 610
034D 611
034E 612
034F 613
0350 614

NOTE: WHEN AT BEGINNING OF BUFFER (NULL PRECEDING CURRENT RECORD),
SPECIAL PROCESSING IS DONE. THE DCL$GET_CURR_COMMAND IS USED
INSTEAD OF DCL$GET_NEXT_COMMAND. THE WRK_RECALL_POINTER IS USED
AS A FLAG TO DETERMINE WHETHER OR NOT THE FIRST COMMAND IN THE
BUFFER HAS BEEN ALREADY PROCESSED. NORMALLY, THE POINTER IS POINTING
TO THE BEGINNING OF A COMMAND (FIRST BYTE ALWAYS NULL). IF THE
POINTER IS POINTING TO A NON-NULL BYTE, IT MEANS THAT THE FIRST
COMMAND IN THE BUFFER HAS ALREADY BEEN PROCESSED (THE POINTER WILL
BE POINTING TO THE LENGTH BYTE FOLLOWING THE NULL).
IN THIS CASE, THE POINTER IS CORRECTED AND THE COMMAND IS
RETRIEVED WITH THE DCL$GET_NEXT_COMMAND INSTEAD OF DCL$GET_CURR_COMMAND.

0280 30 032E 597 RECALL_NEXT:
C5 AA 95 0331 598 BSBW DCL$LOCKED STATE ;RESTORE LOCKED STATE
EC 13 0334 599 TSTB WRK_B_RECALLCNT(R10) ;ANY NEXT COMMANDS TO RECALL?
0336 600 BEQL END_OF_LIST ;NO, THEN REPEAT CURRENT COMMAND
EA BA 95 0336 601 TSTB @WRK_L_RECALLPTR(R10) ;POINTING TO BEGINNING OF COMMAND?
05 13 0339 602 BEQL 25$ ;YES, HANDLE NORMALLY
033B 603
033C 604
033D 605
033E 606
033F 607
0340 608
0341 609
0342 610
0343 611
0344 612
0345 613
0346 614
0347 615
0348 616
0349 617
034A 618
034B 619
034C 620
034D 621
034E 622
034F 623
0350 624

50 EA AA 01 C3 0340 607 25$: SUBL3 #1,WRK_L_RECALLPTR(R10),R0 ;POINT R0 TO CURR. COMMAND-1
51 0133 CB 9E 0345 608 MOVAB PRC_G_COMMANDS(R11),R1 ;R1 = ADDR. OF BEGINNING OF BUF.
51 50 D1 034A 609 CMPL R0,R1 ;BUFFER UNDERFLOW?
05 05 1E 034D 610 BGEQU 28$ ;NO, R0 OK AS IS
50 0401 C0 9E 034F 611 MOVAB PRC_C_CMDRUFISZ(R0),R0 ;YES, POINT R0 TO TOP OF BUFFER
60 95 0354 612 28$: TSTB (R0) ;PREVIOUS COMMAND EXIST?
14 12 0356 613 BNEQ 30$ ;YES, PROCESS NORMALLY
0358 614
```



0358 615 : SPECIAL CASE: THERE ARE EXACTLY 21 COMMANDS IN THE COMMAND BUFFER  
0358 616 : AND RECALL\_PREV HAS RETRIEVED THE LAST VALID COMMAND (IT LEAVES  
0358 617 : POINTER TO FIRST COMMAND.) IN THIS CASE, DO A GET NEXT INSTEAD OF  
0358 618 : GET CURRENT.  
0358 619 :  
C5 AA 91 0358 620 : CMPB WRK\_B\_RECALLCNT(R10),- ;HAVE MAX. COMMANDS BEEN DISPL?  
15 0358 621 : #WRK\_C\_RECALLMAX+1  
OE 13 035C 622 : BEQL 30\$ ;YES, GET NEXT COMMAND.  
035E 623 :  
7E 7C 035E 624 : CLRQ -(SP) ;NO, ALLOCATE A DESCRIPTOR  
SE DD 0360 625 : PUSHL SP ;PUSH DESCRIPTOR ADDR.  
0000'CF 01 FB 0362 626 : CALLS #1,DCL\$GET\_CURR\_COMMAND ;GET CURRENT COMMAND  
EA AA D6 0367 627 : INCL WRK\_L\_RECALLPTR(R10) ;INDICATE 1ST COMMAND PROCESSED.  
A9 11 036A 628 : BRB RECALL\_CURR ;OUTPUT CURRENT COMMAND  
036C 629 :  
7E 7C 036C 630 30\$: CLRQ -(SP) ;ALLOCATE A DESCRIPTOR  
SE DD 036E 631 : PUSHL SP ;PUSH DESCR ADDRESS  
0000'CF 01 FB 0370 632 : CALLS #1,DCL\$GET\_NEXT\_COMMAND ;RECALL THE SPECIFIED COMMAND  
C5 AA 97 0375 633 : DECB WRK\_B\_RECALLCNT(R10) ;DECREMENT THE RECALL COUNT  
9B 12 0378 634 : BNEQ RECALL\_CURR ;BRANCH IF NOT NOW ZERO  
SE 08 C0 037A 635 : ADDL #8,SP ;RESTORE THE STACK  
A3 11 037D 636 : BRB END\_OF\_LIST ;OUTPUT A BLANK LINE

|          |          |         |  |      |     |   |   |  |                                  |
|----------|----------|---------|--|------|-----|---|---|--|----------------------------------|
|          |          |         |  | 037F | 638 | : |   |  |                                  |
|          |          |         |  | 037F | 639 | : | ERASE THE CURRENT LINE IF THE TERMINAL IS ANSI CRT. |  |                                  |
|          |          |         |  | 037F | 640 | : |   |  |                                  |
|          |          |         |  | 037F | 641 | : | RO,R5 ARE DESTROYED                                 |  |                                  |
|          |          |         |  | 037F | 642 | : |   |  |                                  |
|          |          |         |  | 037F | 643 | : | ERASE_LINE:   |  |                                  |
|          |          |         |  | 037F | 644 | : | CLRQ -(SP)  |  | :CREATE ITEM LIST                |
|          |          |         |  | 0381 | 645 | : | PUSHAB -8(SP)                                       |  | :SET BUFFER ADDRESS              |
| 001C0004 | F8 AE    | 7C 9F   |  | 0384 | 646 | : | PUSHL #DVI\$_DEVDEPEND2@16+4                        |  | :SET ITEM CODE                   |
|          | 8F DD    | DD 7C   |  | 038A | 647 | : | CLRQ -(SP)  |  | :ALLOCATE AN IOSB                |
| 50       | 7E SE    | 7C DO   |  | 038C | 648 | : | MOVL SP,R0  |  | :GET ADDRESS OF IOSB             |
|          |          |         |  | 038F | 649 | : | \$GETDVIEW S EFN=#EXESC SYSEFN,-                    |  | :GET DEVDEPEND2                  |
|          |          |         |  | 038F | 650 | : | CHAN=PRC W INPCHAN(R11),-                           |  |                                  |
|          |          |         |  | 038F | 651 | : | ITMLST=8(R0),-                                      |  |                                  |
|          |          |         |  | 038F | 652 | : | IOSB=(R0)   |  |                                  |
|          |          |         |  | 03AB | 653 | : | BLBC R0,90\$  |  | :BRANCH IF ERROR                 |
|          |          |         |  | 03AE | 654 | : | MOVZWL (SP),R0                                      |  | :GET IOSB STATUS                 |
|          |          |         |  | 03B1 | 655 | : | BLBC R0,90\$  |  | :BRANCH IF ERROR                 |
|          |          |         |  | 03B4 | 656 | : |   |  |                                  |
|          |          |         |  | 03B4 | 657 | : | BBC #TT2\$V ANSICRT,8(SP),90\$                      |  | :BRANCH IF ANSI CRT BIT CLEAR    |
|          |          |         |  | 03B9 | 658 | : | MOVL PRC L OUTRAB(R11),R5                           |  | :SET ADDRESS OF OUTPUT FILE RAB  |
| 28 A5    | 55 OC AB | DO 9E   |  | 03BD | 659 | : | MOVAB ERASE%1,RAB\$L_RBF(R5)                        |  | :SET ADDRESS OF OUTPUT RECORD    |
| 22 A5    | FC40 CF  | 9B 03C3 |  | 03C3 | 660 | : | MOVZBW ERASE,RAB\$W_RSZ(R5)                         |  | :SET SIZE OF OUTPUT RECORD       |
|          |          |         |  | 03C9 | 661 | : | DISABLE   |  | :DISABLE CONTROL Y/C AST'S       |
|          |          |         |  | 03CD | 662 | : | INSV #0,#RAB\$V_PPF_RAT,-                           |  | :DISABLE CR FORMAT WRITES        |
|          |          |         |  | 03D0 | 663 | : | #RAB\$\$_PPF_RAT,RAB\$W_ISI(R5)                     |  |                                  |
|          |          |         |  | 03D3 | 664 | : | \$PUT RAB=(R5)                                      |  | :OUTPUT RECORD                   |
|          |          |         |  | 03DC | 665 | : | INSV #FAB\$M_CR,#RAB\$V_PPF_RAT,-                   |  | :ENABLE CR FORMAT WRITES         |
|          |          |         |  | 03DF | 666 | : | #RAB\$\$_PPF_RAT,RAB\$W_ISI(R5)                     |  |                                  |
|          |          |         |  | 03E2 | 667 | : | ENABLE  |  | :ENABLE CONTROL Y/C AST'S        |
|          |          |         |  | 03E4 | 668 | : |   |  |                                  |
|          |          |         |  | 03E4 | 669 | : | MOVL PRC L TRMLIST(R11),R0                          |  | :GET ADDRESS OF XABTRM ITEM LIST |
|          |          |         |  | 03E8 | 670 | : | CLRW @ITRM_L_PMPTADDR(R0)                           |  | :REMOVE CR/LF FROM PROMPT STRING |
|          |          |         |  | 03EB | 671 | : |   |  |                                  |
|          |          |         |  | 03EB | 672 | : | ADDL #6*4,SP  |  | :RESTORE STACK                   |
|          |          |         |  | 03EE | 673 | : | RSB   |  |                                  |

```

                                03EF 675 :
                                03EF 676 : REPROMPT WITH RECALLED COMMAND.
                                03EF 677 :
                                03EF 678 INSERT_COMMAND:
50 10 AB D0 03EF 679      MOVL PRC L TRMLIST(R11),R0      ;GET ADDRESS OF XABTRM ITEM LIST
    28 A0 D4 03F3 680      CLRL ITRM_C_OFFSET(R0)        ;REQUEST A FRESH READ
18 A0 51 B0 03F6 681      MOVW R1,ITRM_W_INILEN(R0)       ;SET LENGTH OF INITIAL STRING
1C A0 52 D0 03FA 682      MOVL R2,ITRM_L_INIADDR(R0)      ;SET ADDRESS OF INITIAL STRING
50 40 A4 D0 03FE 683      MOVL RAB$XABTR4),R0             ;GET ADDRESS OF XABTRM
    30 D0 0402 684      MOVL #ITRM_K_LENGTH,-             ;SET EXPANDED LENGTH OF ITEM LIST
    OC A0 D0 0404 685      MOVL XAB$W_ITMLST_LEN(R0)      ;
    05 0406 686      RSB                                ;RETURN

```



```
0407 688 .SBTTL PROCESS ESCAPE SEQUENCES
0407 689 ---
0407 690 PROCESS_ESCAPE - PROCESS ESCAPE SEQUENCES
0407 691
0407 692 SUBROUTINE TO CHECK FOR AND PROCESS ESCAPE SEQUENCES.
0407 693
0407 694 INPUTS:
0407 695
0407 696 R2 = ADDRESS OF NEW INPUT RECORD
0407 697 R3 = LENGTH OF NEW INPUT RECORD, EXCLUDING NULL AT END OF LINE
0407 698 R4 = ADDRESS OF RAB
0407 699
0407 700 OUTPUTS:
0407 701
0407 702 R2 = ADDRESS OF NEW INPUT RECORD
0407 703 R3 = LENGTH OF NEW INPUT RECORD, EXCLUDING NULL AT END OF LINE
0407 704 ---
0407 705
0407 706 PROCESS_ESCAPE:                                ;PROCESS ESCAPE SEQUENCES
0407 707
0407 708 :
0407 709 : CHECK FOR ESCAPE SEQUENCES.
0407 710 :
0407 711 : CMPB #1,RAB$W_STV2(R4)                                ;ESCAPE SEQUENCE?
0407 712 : BLSSU 5$                                           ;YES, THEN PROCESS
0407 713 : BSBW DCL$LOCKED_STATE                            ;RESTORE LOCKED KEYPAD STATE
0407 714 : MOVL #1,R0                                     ;SET REPROMPT STATUS
0407 715 : RSB                                           ;RETURN
0407 716
0407 717 :
0407 718 : REPROMPT IF NO KEY OR SYMBOL FOUND.
0407 719 :
0407 720 70$: MOVQ (SP)+,R4                                ;RESTORE R4/R5
0407 721 80$: MOVQ (SP)+,R2                                ;RESTORE R2/R3
0407 722 : MOVL R3,R1                                     ;CREATE R1/R2 DESCRIPTOR
0407 723 : BSBW INSERT_COMMAND                            ;REPROMPT WITH THIS COMMAND
0407 724 : BSBW DCL$LOCKED_STATE                        ;RESTORE LOCKED KEYPAD STATE
0407 725 : MOVL #3,R0                                     ;SET REPROMPT STATUS
0407 726 : RSB                                           ;RETURN
0407 727
0407 728 :
0407 729 : FIND ASSOCIATED META-KEY NAME.
0407 730 :
0407 731 5$: MOVQ R2,-(SP)                                ;SAVE R2/R3
0407 732 : CLRL -(SP)                                       ;ALLOCATE SPACE FOR RETURN ADDRESS
0407 733 : PUSHL SP                                         ;POINT AT IT
0407 734 : MOVZBL RAB$W_STV2(R4),-(SP)                  ;GET TERMINATOR LENGTH
0407 735 : PUSHAB (R2)[R3]                                ;GET TERMINATOR ADDRESS
0407 736 : CALLS #3,G^GET_KEY_NAME                        ;LOOK UP THE ESCAPE SEQUENCE
0407 737 : MOVL (SP)+,R2                                  ;GET ADDRESS OF ASCII META-KEY NAME
0407 738 : BLBC R0,80$                                    ;SKIP IF NONE
0407 739
0407 740 :
0407 741 : SPECIAL CASE UP AND DOWN ARROW KEYS.
0407 742 :
0407 743 :
0407 744 : CMPB (R2),#4                                ;IS THE STRING SHORT ENOUGH?
0407 745 : BGTR 10$                                       ;NO, THEN SKIP
```

0E A4 01 91 0407 711  
19 1F 0408 712  
01A1 30 040D 713  
50 01 D0 0410 714  
05 0413 715  
0414 716  
0414 717  
0414 718  
0414 719  
54 8E 7D 0414 720 70\$: MOVQ (SP)+,R4  
52 8E 7D 0417 721 80\$: MOVQ (SP)+,R2  
51 53 D0 041A 722 : MOVL R3,R1  
D0 10 041D 723 : BSBW INSERT\_COMMAND  
018F 30 041F 724 : BSBW DCL\$LOCKED\_STATE  
50 03 D0 0422 725 : MOVL #3,R0  
05 0425 726 : RSB  
0426 727  
0426 728  
0426 729  
0426 730  
7E 52 7D 0426 731 5\$: MOVQ R2,-(SP)  
7E D4 0429 732 : CLRL -(SP)  
5E DD 042B 733 : PUSHL SP  
7E 0E A4 9A 042D 734 : MOVZBL RAB\$W\_STV2(R4),-(SP)  
6243 9F 0431 735 : PUSHAB (R2)[R3]  
00000000 GF 03 FB 0434 736 : CALLS #3,G^GET\_KEY\_NAME  
52 8E D0 043B 737 : MOVL (SP)+,R2  
D6 50 E9 043E 738 : BLBC R0,80\$  
0441 739  
0441 740  
0441 741  
0441 742  
04 62 91 0441 743 : CMPB (R2),#4  
1E 14 0444 744 : BGTR 10\$

```
5055 8F 01 A2 B1 0446 745      CMPW 1(R2),#*A'UP'      ;CHECK FOR UP ARROW
10      13 044C 746      BEQL 8$      ;BRANCH IF MATCH
4E574F44 8F 01 A2 D1 044E 747      CMPL 1(R2),#*A'DOWN'    ;CHECK FOR DOWN ARROW
OC      12 0456 748      BNEQ 10$     ;BRANCH IF NO MATCH
52      8E 7D 0458 749 7$:      MOVQ (SP)+,R2      ;RESTORE R2/R3
FED0    31 045B 750      BRW RECALL_NEXT      ;DO THE RECALL
52      8E 7D 045E 751 8$:      MOVQ (SP)+,R2      ;RESTORE R2/R3
FE6B    31 0461 752      BRW RECALL_PREV      ;DO THE RECALL
0464 753
0464 754
0464 755      : FIND ASSOCIATED SYMBOL VALUE.
0464 756      (SP) = INITIAL OFFSET
0464 757      4(SP) = NEW COMMAND LENGTH
0464 758      8(SP) = RAB ADDRESS
0464 759      12(SP) = R5
0464 760      16(SP) = INITIAL COMMAND ADDRESS
0464 761      20(SP) = INITIAL (LATER NEW) COMMAND LENGTH
0464 762
0464 763 10$:      MOVQ R4,-(SP)      ;SAVE R4/R5
51      82 9A 0467 765      MOVZBL (R2)+,R1      ;GET DESCRIPTOR OF SYMBOL NAME
FB93    30 046A 766      BSBW DCL$SEARCH_KEYPAD      ;SEARCH SYMBOL TABLE FOR MATCH
A4 50    E9 046D 767      BLBC R0,70$      ;SKIP IF NOT FOUND
0470 768
0470 769
0470 770      : /ERASE - COPY STRING INTO INPUT BUFFER.
0470 771
2A 54    04 E1 0470 772      BBC #SYM_V_ERASE,R4,20$      ;IF /NOERASE, THEN SKIP
51      DD 0474 773      PUSHL R1      ;SAVE NEW LENGTH
7E      D4 0476 774      CLRL -(SP)      ;SET INITIAL OFFSET
55      10 AE D0 0478 775      MOVL 16(SP),R5      ;GET INITIAL ADDRESS
54      DD 047C 776      PUSHL R4      ;SAVE FLAGS
65 62    51 28 047E 777      MOVCL R1,(R2),(R5)      ;COPY VALUE TO INPUT BUFFER
54      8E D0 0482 778      POPL R4      ;RESTORE FLAGS
4D      11 0485 779      BRB 30$      :
0487 780
0487 781 95$:      STATUS BUFOVF      ;SET OVERFLOW STATUS
0120    30 048E 782      BSBW DCL$LOCKED_STATE      ;RESTORE LOCKED KEYPAD STATE
00E0    31 0491 783      BRW 67$      ;RETURN ERROR
0494 784
0494 785 96$:      STATUS SYMOVF      ;SET OVERFLOW STATUS
00D4    31 0498 786      BRW 66$      ;RETURN
049E 787
049E 788
049E 789      : /NOERASE - COPY STRING INTO INPUT BUFFER.
049E 790
7E      OC AE 51 C1 049E 791 20$:      ADDL3 R1,12(SP),-(SP)      ;SAVE NEW LENGTH
00000100 8F 6E D1 04A3 792      CMPL (SP),#WRK_C_INPBUFSIZ      ;WILL STRING FIT IN BUFFER?
DB      1A 04AA 793      BGTRJ 95$      ;NO, THEN RETURN ERROR
55      OC AE 10 AE C1 04AC 794      ADDL3 16(SP),12(SP),R5      ;FIND CURRENT EOL
50      04 AE D0 04B2 795      MOVL 4(SP),R0      ;GET ADDRESS OF RAB
53      OF A0 9A 04B6 796      MOVZBL RAB$W_STV2+1(R0),R3      ;GET CURSOR OFFSET FROM EOL
55      53 C2 04BA 797      SUBL R3,R5      ;BACK UP TO INSERTION POINT
7E      10 AE 53 C3 04BD 798      SUBL3 R3,16(SP),-(SP)      ;SAVE INITIAL OFFSET
3E      BB 04C2 799      PUSHL #*M<R1,R2,R3,R4,R5>      ;SAVE REGISTERS
6541    65 53 28 04C4 800      MOVCL R3,(R5),(R5)[R1]      ;MOVE THE TEXT
3E      BA 04C9 801      POPR #*M<R1,R2,R3,R4,R5>      ;RESTORE REGISTERS
```

```
65 62 54 DD 04CB 802 PUSHI R4
51 28 04CD 803 MOVCS R1,(R2),(R5)
54 8ED0 04D1 804 POPL R4
04D4 805
04D4 806
04D4 807
04D4 808
04D4 809 30$: BSBW DCL$LOCKED STATE
E1 04D7 810 BBC #SYM_V_STATE,R4,50$
52 51 D0 04DB 811 MOVL R1,R2
51 82 9A 04DE 812 MOVZBL (R2)+,R1
FB1C 30 04E1 813 BSBW DCL$ALLOC_STATE
AD 50 E9 04E4 814 BLBC R0,96$
OC 54 03 E1 04E7 815 BBC #SYM_V_LOCK,R0,3$
50 4C AB D0 04EB 816 MOVL PRC [LASTKEY(R11),R0
FB0E 30 04EF 817 BSBW DCL$DEALLOC_STATE
48 AB D0 04F2 818 MOVL PRC [CURRKEY(R11),-
4C AB 04F5 819 PRC [LASTKEY(R11)
04F7 820
04F7 821
04F7 822
04F7 823
04F7 824 50$: BBC #SYM_V_ERASE,R4,52$
FE81 30 04FB 825 BSBW ERASE [LINE
2D 54 01 E1 04FE 826 BBC #SYM_V_TERMINATE,R4,52$
55 OC AB D0 0502 827 MOVL PRC [OUTRAB(R11),R5
50 10 AB D0 0506 828 51$: MOVL PRC [TRMLIST(R11),R0
10 A0 D0 050A 829 MOVL ITRM [PMPTADDR(R0),-
28 A5 050D 830 RAB$ [RBF(R5)
OC A0 B0 050F 831 ITRM [PMPTLEN(R0),-
22 A5 0512 832 RAB$ [RSZ(R5)
0514 833
0518 834 DISABLE
02 A5 08 F0 051B 835 INSV #0,#RAB$V_PPF_RAT,-
051E 836 #RAB$S_PPF_RAT,RAB$W_ISI(R5)
06 02 F0 0527 837 $PUT RAB=(R5)
02 A5 08 052A 838 INSV #FAB$M_CR,#RAB$V_PPF_RAT,-
052D 839 #RAB$S_PPF_RAT,RAB$W_ISI(R5)
052F 840
052F 841
052F 842
052F 843
052F 844
052F 845
052F 846
052F 847
052F 848
052F 849
052F 850
052F 851
052F 852
052F 853
052F 854
052F 855
052F 856
052F 857
052F 858
052F 859
052F 860
052F 861
052F 862
052F 863
052F 864
052F 865
052F 866
052F 867
052F 868
052F 869
052F 870
052F 871
052F 872
052F 873
052F 874
052F 875
052F 876
052F 877
052F 878
052F 879
052F 880
052F 881
052F 882
052F 883
052F 884
052F 885
052F 886
052F 887
052F 888
052F 889
052F 890
052F 891
052F 892
052F 893
052F 894
052F 895
052F 896
052F 897
052F 898
052F 899
052F 900
052F 901
052F 902
052F 903
052F 904
052F 905
052F 906
052F 907
052F 908
052F 909
052F 910
052F 911
052F 912
052F 913
052F 914
052F 915
052F 916
052F 917
052F 918
052F 919
052F 920
052F 921
052F 922
052F 923
052F 924
052F 925
052F 926
052F 927
052F 928
052F 929
052F 930
052F 931
052F 932
052F 933
052F 934
052F 935
052F 936
052F 937
052F 938
052F 939
052F 940
052F 941
052F 942
052F 943
052F 944
052F 945
052F 946
052F 947
052F 948
052F 949
052F 950
052F 951
052F 952
052F 953
052F 954
052F 955
052F 956
052F 957
052F 958
052F 959
052F 960
052F 961
052F 962
052F 963
052F 964
052F 965
052F 966
052F 967
052F 968
052F 969
052F 970
052F 971
052F 972
052F 973
052F 974
052F 975
052F 976
052F 977
052F 978
052F 979
052F 980
052F 981
052F 982
052F 983
052F 984
052F 985
052F 986
052F 987
052F 988
052F 989
052F 990
052F 991
052F 992
052F 993
052F 994
052F 995
052F 996
052F 997
052F 998
052F 999
052F 1000
```

|    |    |    |      |     |        |      |                          |                                    |
|----|----|----|------|-----|--------|------|--------------------------|------------------------------------|
|    |    |    | 056D | 859 | ENABLE |      |                          |                                    |
|    |    |    | 056F | 860 |        |      |                          | :ENABLE CONTROL Y/C AST'S          |
| 50 | 01 | D0 | 056F | 861 | 65\$:  | MOVL | #1,R0                    | :SET NORMAL RETURN STATUS          |
|    | 8E | D5 | 0572 | 862 | 66\$:  | TSTL | (SP)+                    | :POP INITIAL OFFSET                |
| 0C | AE | D0 | 0574 | 863 | 67\$:  | MOVL | (SP)+,12(SP)             | :REPLACE OLD LENGTH WITH NEW       |
| 54 | 8E | 7D | 0578 | 864 |        | MOVQ | (SP)+,R4                 | :RESTORE R4/R5                     |
| 52 | 8E | 7D | 057B | 865 |        | MOVQ | (SP)+,R2                 | :RESTORE R2/R3                     |
|    |    | 05 | 057E | 866 |        | RSB  |                          | :RETURN                            |
|    |    |    | 057F | 867 |        |      |                          |                                    |
|    |    |    | 057F | 868 |        |      |                          |                                    |
|    |    |    | 057F | 869 |        |      |                          |                                    |
|    |    |    | 057F | 870 |        |      |                          |                                    |
|    |    |    | 057F | 871 | 60\$:  | MOVL | PRC_L_TRMLIST(R11),R0    | :GET ADDRESS OF XABTRM ITEM LIST   |
| 50 | 10 | D0 | 057F | 871 |        | MOVL | (SP)+,ITRM_L_OFFSET(R0)  | :SET CHARACTER TO START ECHOING AT |
| 28 | A0 | D0 | 0583 | 872 |        | INCL | ITRM_L_OFFSET(R0)        |                                    |
|    | 28 | D6 | 0587 | 873 |        | BBC  | #SYM_V_ERASE,R4,61\$     | :IF /NOERASE, THEN SKIP            |
| 03 | 54 | E1 | 058A | 874 |        | CLRL | ITRM_L_OFFSET(R0)        | :START WITH FIRST CHARACTER        |
|    | 28 | D4 | 058E | 875 | 61\$:  | MOVL | (SP)+,T2(SP)             | :REPLACE OLD LENGTH WITH NEW       |
| 0C | AE | D0 | 0591 | 876 |        | MOVW | 12(SP),ITRM_W_INILEN(R0) | :SET LENGTH OF INITIAL STRING      |
| 18 | A0 | B0 | 0595 | 877 |        | MOVL | 8(SP),ITRM_L_INIADDR(R0) | :SET ADDRESS OF INITIAL STRING     |
| 1C | A0 | D0 | 059A | 878 |        | MOVQ | (SP)+,R4                 | :RESTORE R4/R5                     |
|    | 54 | 7D | 059F | 879 |        | MOVL | RAB\$L_XAB(R4),R0        | :GET ADDRESS OF XABTRM             |
| 50 | 40 | D0 | 05A2 | 880 |        | MOVL | #ITRM_K_LENGTH,-         | :SET EXPANDED LENGTH OF ITEM LIST  |
|    | 30 | D0 | 05A6 | 881 |        |      | XAB\$W_ITMLST_LEN(R0)    |                                    |
|    | 0C |    | 05A8 | 882 |        |      |                          |                                    |
|    | 8E | 7D | 05AA | 883 |        | MOVQ | (SP)+,R2                 | :RESTORE R2/R3                     |
| 52 | 8E | D0 | 05AD | 884 |        | MOVL | #3,R0                    | :SET REPROMPT RETURN STATUS        |
| 50 | 03 | D0 | 05AD | 884 |        | RSB  |                          | :RETURN                            |
|    |    | 05 | 05B0 | 885 |        |      |                          |                                    |



```

0581 887 .SBTTL RESTORE LOCKED KEYPAD STATE
0581 888 ---
0581 889 DCL$LOCKED_STATE - RESTORE LOCKED KEYPAD STATE
0581 890
0581 891 DELETE ANY TEMPORARY KEYPAD STATE THAT MAY HAVE BEEN SET AND RESTORE THE
0581 892 LOCKED KEYPAD STATE.
0581 893
0581 894 INPUTS:
0581 895
0581 896 R11 = ADDRESS OF PRC DATA STRUCTURE
0581 897 PRC_L_CURRKEY = CURRENT, POSSIBLY TEMPORARY, KEYPAD STATE
0581 898 PRC_L_LASTKEY = DEFAULT LOCKED KEYPAD STATE
0581 899
0581 900 OUTPUTS:
0581 901
0581 902 PRC_L_CURRKEY IS UPDATED.
0581 903
0581 904 ---
0581 905
0581 906 DCL$LOCKED_STATE::
0581 907 PUSH R0
0583 908 CMPL PRC_L_CURRKEY(R11),-
0586 909 PRC_L_LASTKEY(R11)
0588 910 BEQL 90$
058A 911 MOVL PRC_L_CURRKEY(R11),R0
058E 912 BSBW DCL$DEALLOC_STATE
05C1 913 MOVL PRC_L_LASTKEY(R11),-
05C4 914 PRC_L_CURRKEY(R11)
05C6 915 90$: POPL R0
05C9 916 RSB
05CA 917

50 50 DD 0581 907
48 AB D1 0583 908
4C AB 0586 909
OC 13 0588 910
50 48 AB D0 058A 911
FA3F' 30 058E 912
4C AB D0 05C1 913
48 AB 05C4 914
50 BED0 05C6 915
OS 05C9 916
05CA 917

;RESTORE LOCKED KEYPAD STATE
;SAVE R0
;IS TEMPORARY STATE IN EFFECT?
;
;NO, THEN DONE
;GET ADDRESS OF ASCIC STATE
;DEALLOCATE TEMPORARY STATE
;RESTORE THE LOCKED STATE
;
;RESTORE R0
;RETURN

```

```
05CA 919 .SBTTL EXPAND INPUT LINE
05CA 920 ---
05CA 921 EXPAND - EXPAND INPUT LINE WITH SYMBOL SUBSTITUTIONS
05CA 922
05CA 923 SUBROUTINE TO EXPAND INPUT LINE BY EXECUTING ALL STRING SUBSTITUTION
05CA 924 COMMANDS. THE UNUSED AREA IN THE EXPANSION BUFFER IS USED TEMPORARILY
05CA 925 TO HOLD THE EXPANDED COPY OF THE COMMAND LINE. AFTER ALL SUBSTITUTIONS
05CA 926 ARE PERFORMED, THE EXPANDED COPY IS COPIED BACK INTO THE INPUT RECORD
05CA 927 BUFFER SO THAT SEMANTIC PARSING CAN CONTINUE.
05CA 928
05CA 929 INPUTS:
05CA 930
05CA 931 WRK_L_CHARPTR = POINTER TO NEW INPUT RECORD
05CA 932
05CA 933 OUTPUTS:
05CA 934
05CA 935 R2 = ADDRESS OF NEW INPUT RECORD
05CA 936 R3 = LENGTH OF NEW INPUT RECORD, EXCLUDING NULL AT END OF LINE
05CA 937 WRK_L_CHARPTR = POINTER TO EXPANDED INPUT RECORD
05CA 938 ---
05CA 939
05CA 940 EXPAND:
05CA 941 PUSHF #M<R4,R5,R6,R7,R8,R9> :EXPAND INPUT LINE
05CA 942 PUSHF WRK_L_MARKPTR(R10) :SAVE REGISTERS
05CA 943 PUSHF WRK_L_EXPANDPTR(R10) :SAVE MARKER POINTER
05CA 944 MOVW WRK_W_FLAGS(R10),-(SP) :SAVE EXPANSION BUFFER POINTER
05CA 945 BISW #WRK_M_INPSUBST!WRK_M_NOUPCASE!WRK_M_STAR,WRK_W_FLAGS(R10) :SAVE CURRENT PARSING FLAGS
05CA 946 :PREVENT PROCESSING OF T,-,Q,ETC.
05CA 947 :AND PREVENT UPCASING OF INPUT CHARS
05CA 948 :AND ACCEPT '*' AS A TERMINATOR
05CA 949 CLRW -(SP) :INITIALIZE ITERATION COUNTER
05CA 950
05CA 951 GET NEXT CHARACTER FROM INPUT RECORD
05CA 952
05CA 953 10$: BSBW DCL$GETCHAR :GET NEXT CHARACTER FROM INPUT LINE
05CA 954 CMPB RO,#A/'/ :STRING SUBSTITUTION COMMAND?
05CA 955 BNEQ 70$ :IF NEQ NO
05CA 956 BBC #WRK_V_QUOTE,WRK_W_FLAGS(R10),20$ :IF LBC NOT IN QUOTE
05CA 957 BSBW DCL$SETCHAR :CHECK NEXT CHARACTER
05CA 958 CMPB RO,#A/'/ :NEXT CHARACTER ALSO SINGLE QUOTE?
05CA 959 BNEQ 60$ :IF NEQ NO
05CA 960 BSBW DCL$GETCHAR :GOBBLE SECOND SINGLE QUOTE
05CA 961 BRB 20$ :TRY SYMBOL SUBSTITUTION
05CA 962
05CA 963 : ONE SINGLE QUOTE WAS DETECTED IN A DOUBLE QUOTED STRING - TREAT LITERALLY
05CA 964
05CA 965 60$: MOVZBL #A/'/,RO :INSERT SINGLE QUOTE WITHIN STRING
05CA 966
05CA 967 : IF COMMENT IS DETECTED, THEN DO SUBSTITUTIONS BUT IF AN ERROR IS
05CA 968 : DETECTED, THEN NO ERROR IS ISSUED AND A NULL STRING IS SUBSTITUTED.
05CA 969
05CA 970 70$: BBS #WRK_V_QUOTE,WRK_W_FLAGS(R10),80$ :BRANCH IF IN QUOTED STRING
05CA 971 CMPB RO,#A'!' :START OF COMMENT STRING?
05CA 972 BNEQ 80$ :BRANCH IF NOT A COMMENT
05CA 973 SETBIT WRK_V_COMMENT,WRK_W_FLAGS(R10) :MARK WE ARE IN A COMMENT
05CA 974 :SO THAT ERRORS ARE NOT REPORTED
05CA 975 :
```

03FO 8F BB 05CA 941  
F48A CA DD 05CE 942  
F486 CA DD 05D2 943  
7E FO AA BO 05D6 944  
FO AA OC20 8F AB 05DA 945  
05E0 946  
05E0 947  
05E0 948  
7E B4 05E0 949  
05E2 950  
05E2 951  
05E2 952  
27 FA1B' 30 05E2 953  
27 50 91 05E5 954  
15 12 05E8 955  
27 FO AA 04 E1 05EA 956  
FA0E' 30 05EF 957  
27 50 91 05F2 958  
05 12 05F5 959  
FA06' 30 05F7 960  
1A 11 05FA 961  
05FC 962  
05FC 963  
05FC 964  
50 27 9A 05FC 965  
05FF 966  
05FF 967  
05FF 968  
05FF 969  
OA FO AA 04 E0 05FF 970  
21 50 91 0604 971  
05 12 0607 972  
0609 973  
060E 974  
060E 975

```
060E 976 : WRITE THE CURRENT CHARACTER INTO THE EXPANSION BUFFER AND LOOP
060E 977 :
F9EF' 30 060E 978 80$: BSBW DCL$PUTCHAR :PUT CHARACTER IN EXPANSION BUFFER
CF 12 0611 979 : BNEQ 10$ :IF NOT EOL, KEEP SCANNING
00B9 31 0613 980 : BRW 90$ :END OF LINE - TERMINATE SCAN
0616 981 :
0616 982 : SYMBOL SUBSTITUTION REQUESTED. GET THE SYMBOL AND SEARCH THE SYMBOL
0616 983 : TABLE, AND IF NOT FOUND THERE, TRY AS A LEXICAL FUNCTION.
0616 984 :
F486 CA DD 0616 985 20$: PUSHL WRK_L_EXPANDPTR(R10) :SAVE PLACE IN EXPANSION BUFFER
7E FO AA BO 061A 986 : MOVW WRK_W_FLAGS(R10),-(SP) :SAVE LEXICAL FLAGS
FO AA 10 AA 061E 987 : BICW #WRK_M_QUOTE,WRK_W_FLAGS(R10)
OC EO 0622 988 : BBS #WRK_V_COMMENT,- :ARE WE IN A COMMENT?
06 FO AA 0624 989 : WRK_W_FLAGS(R10),22$
FO AA 0800 8F AA 0627 990 : BICW #WRK_M_NOUPCASE,WRK_W_FLAGS(R10) :NO, THEN UPCASE THE SYMBOL
062D 991 : :PRETEND WE'RE NOT IN A STRING
062D 992 : :SO GETOKEN STOPS BEFORE END-OF-STRING
50 27 9A 062D 993 22$: MOVZBL #A/'/',R0 :INSERT SINGLE QUOTE WITHIN STRING
F9CD' 30 0630 994 : BSBW DCL$PUTCHAR :IN CASE SUBSTITUTION NOT ALLOWED
F9CA' 30 0633 995 : BSBW DCL$GETOKEN :GET/COPY NEXT TOKEN
0636 996 :
0636 997 : IF IN COMMENT, ONLY ALLOW F$VERIFY TO BE SUBSTITUTED
0636 998 :
29 FO AA OC E1 0636 999 : BBC #WRK_V_COMMENT,WRK_W_FLAGS(R10),28$ :BRANCH IF NOT IN COMMENT
04 51 D1 063B 1000 : CMPL R1,#4 :AT LEAST 4 CHARACTER TOKEN?
1C 19 063E 1001 : BLSS 25$ :IF NOT, SKIP SYMBOL SUBSTITUTION
7E 62 D0 0640 1002 : MOVL (R2),-(SP) :PUSH FIRST FOUR CHARACTERS
7E 51 7D 0643 1003 : MOVQ R1, -(SP) :SAVE DESCRIPTOR
51 04 9A 0646 1004 : MOVZBL #4,R1 :CREATE TEMPORARY DESCRIPTOR
52 08 AE 9E 0649 1005 : MOVAB 8(SP),R2
F980' 30 064D 1006 : BSBW DCL$UPCASE :UPCASE THE SYMBOL
51 8E 7D 0650 1007 : MOVQ (SP)+,R1 :RESTORE THE REGISTERS
45562446 8F 8E D1 0653 1008 : CMPL (SP)+,#A'F$VE' :IS IT F$VERIFY WITHIN A COMMENT?
08 13 065A 1009 : BEQL 28$ :IF SO, ALLOW SUBSTITUTION
02 AE F486 CA D0 065C 1010 25$: MOVL WRK_L_EXPANDPTR(R10),2(SP) :COPY 'SYMBOL TO EXPANSION BUFFER
45 11 0662 1011 : BRB 50$
51 D5 0664 1012 28$: TSTL R1 :ZERO LENGTH SYMBOL?
1F 13 0666 1013 : BEQL 40$ :IF EQL YES
56 51 7D 0668 1014 : MOVQ R1,R6 :SAVE STRING PARAMETERS
F992' 30 066B 1015 : BSBW DCL$SEARCH :SEARCH FOR SYMBOL
16 50 E8 066E 1016 : BLBS R0,40$ :IF LBS SYMBOL DEFINITION FOUND
0671 1017 : CLRBIT WRK_V_INPSUBST,WRK_W_FLAGS(R10) :DO PROCESSING OF !,-,@,ETC.
0676 1018 : :TO ALLOW CONTINUATIONS IN FUNCT. ARGS
09 FO AA OC E1 0676 1019 : BBC #WRK_V_COMMENT,- :ARE WE IN A COMMENT?
51 56 7D 0678 1020 : WRK_W_FLAGS(R10),32$
F97F' 30 067E 1021 : MOVQ R6,R1 :YES, THEN UPCASE 'F$VERIFY'
56 51 7D 0681 1022 : BSBW DCL$UPCASE
F979' 30 0684 1023 : MOVQ R1,R6
F976' 30 0687 1024 32$: BSBW DCL$LEXIF :EVALUATE LEXICAL FUNCTION
068A 1025 40$: BSBW DCL$CVT_STRING :CONVERT RESULT TO CHARACTER STRING
068F 1026 : SETBIT WRK_V_INPSUBST,WRK_W_FLAGS(R10) :DISABLE !,-,@,ETC.
068F 1027 :
068F 1028 : TRAILING SINGLE QUOTES ARE OPTIONAL AFTER SYMBOL - GOBBLE IT
068F 1029 :
F96E' 30 068F 1030 : BSBW DCL$SETCHAR :PEEK AT NEXT CHARACTER
27 50 91 0692 1031 : CMPB R0,#A/'/' :SYMBOL END WITH SINGLE QUOTE?
03 12 0695 1032 : BNEQ 30$ :BRANCH IF TRAILING SINGLE QUOTE
```

```
F966' 30 0697 1033 BSBW DCL$GETCHAR ;GOBBLE TRAILING QUOTE
069A 1034
069A 1035 : APPEND THE SYMBOL TRANSLATION TO THE FRONT OF THE INPUT BUFFER
069A 1036 : AND RESET THE INPUT POINTER TO POINT TO IT. THIS IS DONE IN CASE
069A 1037 : THERE ARE ANY SINGLE QUOTES IN THE TRANSLATION WHICH REQUIRE SUBSTITUTION.
069A 1038
F48E CA 51 C2 069A 1039 50$: SUBL R1,WRK_L_CHARPTR(R10) ;CALCULATE ADDRESS TO COPY STRING
50 F48E CA D0 069F 1040 : POINT TO NEW POSITION IN INBUF
01 A0 62 51 28 06A4 1041 : CONCATENATE STRING TO INPUT BUFFER
06A9 1042
06A9 1043 : A MAXIMUM OF 1000 SUBSTITUTIONS IS ALLOWED PER LINE, TO PREVENT ANY
06A9 1044 : INFINITE LOOPS FROM OCCURRING DUE TO RECURSIVE SUBSTITUTIONS.
06A9 1045
F0 AA 8E B0 06A9 1046 50$: MOVW (SP)+,WRK_W_FLAGS(R10) ;RESTORE FLAGS QUOTE AND NOUPCASE
FF28 6E 01 03E8 8F 3D 06AD 1047 : RETRIEVE ADDRESS IN EXPANSION BUFFER
F486 CA 57 56 C1 06B2 1048 ACBW #1000,#1,(SP),10$ ;CHECK FOR SUBSTITUTION LOOP
F48A CA 77 9E 06BA 1049 STATUS EXPSYN ;EXPRESSION SYNTAX ERROR
F931' 31 06C1 1050 ADDL3 R6,R7,WRK_L_EXPANDPTR(R10) ;POINT AT END OF SYMBOL
06C7 1051 MOVAB -(R7),WRK_L_MARKPTR(R10) ;SET ADDRESS OF ""
06CC 1052 BRW DCL$PARSER ;REPORT ERROR
06CF 1053
06CF 1054 : END OF LINE DETECTED. MOVE EXPANDED LINE BACK INTO THE INPUT BUFFER
06CF 1055 : FOR THE LEXICAL PROCESSING.
06CF 1056
57 04 AE D0 06CF 1057 90$: MOVL 4(SP),R7 ;GET SAVED EXPANSION POINTER
59 F486 CA 57 C3 06D3 1058 : CALCULATE LENGTH OF EXPANDED LINE
58 F996 CA 9E 06D9 1059 MOVAB WRK_G_INPBUF+WRK_C_INPBUFSI2(R10),R8 ;FIND END OF INPUT BUFFER
58 59 C2 06DE 1060 : COMPUTE ADDRESS TO MOVE LINE TO
68 67 59 28 06E1 1061 : MOVE EXPANDED LINE TO END OF INPUT BUFFER
52 58 7D 06E5 1062 MOVQ R8,R2 ;SET INPUT LINE PARAMETERS
53 D7 06E8 1063 DECL R3 ;DECREMENT LENGTH TO EXCLUDE EOL CHAR
F48E CA FF A2 9E 06EA 1064 MOVAB -1(R2),WRK_L_CHARPTR(R10) ;SET ADDRESS OF EXPANDED INPUT LINE
8E B5 06F0 1065 TSTW (SP)+ ;REMOVE ITERATION COUNTER FROM STACK
F0 AA 8E B0 06F2 1066 MOVW (SP)+,WRK_W_FLAGS(R10) ;RESTORE CURRENT PARSING FLAGS
F486 CA 8ED0 06F6 1067 POPL WRK_L_EXPANDPTR(R10) ;RESTORE EXPANSION BUFFER POINTER
F48A CA 8ED0 06FB 1068 POPL WRK_L_MARKPTR(R10) ;RESTORE MARKER POINTER
03F0 8F BA 0700 1069 POPR #M<R4,R5,R6,R7,R8,R9> ;RESTORE REGISTERS
05 0704 1070 RSB
```



```

0705 1072 .SBTTL SPECIAL TOKEN LEXICAL PROCESSING
0705 1073 ---
0705 1074 :
0705 1075 : THIS ROUTINE IS CALLED TO PROCESS LEXICAL TOKENS WITH SPECIAL CHARACTERS.
0705 1076 : THE LIST OF CHARACTERS ARE:
0705 1077 :
0705 1078 :     &SYMBOL      - TOKEN IS SUBSTITUTED WITH SYMBOL VALUE
0705 1079 :     @FILESPEC    - TOKEN IS SUBSTITUTED WITH FIRST RECORD CONTAINED
0705 1080 :                   IN THE ASSOCIATED PROCEDURE FILE.
0705 1081 :
0705 1082 : INPUTS:
0705 1083 :
0705 1084 :     R11 = ADDRESS OF PRC AREA
0705 1085 :     R10 = ADDRESS OF WRK AREA
0705 1086 :     RO = FIRST CHARACTER IN TOKEN
0705 1087 :
0705 1088 : ADDITIONAL INPUT AND OUTPUT SPECIFICATIONS ARE GIVEN WITH EACH ROUTINE.
0705 1089 :
0705 1090 : ---
0705 1091 :
0705 1092 DCL$SPECIAL::
0705 1093 :CMPB    RO,#^A'&'      :SUBSTITUTION?
0705 1094 :BEQL     AMPERSAND      :BRANCH IF SO
0705 1095 :CMPB    RO,#^A'@'      :INDIRECTION?
0705 1096 :BEQL     INDIRECT       :BRANCH IF SO
0705 1097 :RSB                      :IF NOT KNOWN, IGNORE IT

```

26 50 91  
40 8F 07 13 0708 1094  
50 91 070A 1095  
22 13 070E 1096  
05 0710 1097

```
0711 1099 .SBTTL PROCESS &SYMBOL CONSTRUCT
0711 1100 ---
0711 1101
0711 1102 HANDLE &SYMBOL CONSTRUCT.
0711 1103
0711 1104 INPUTS:
0711 1105
0711 1106 R0 = CHARACTER IN INPUT BUFFER (&)
0711 1107 R1/R2 = DESCRIPTOR OF TOKEN, INCLUDING '&' CHARACTER
0711 1108
0711 1109 OUTPUTS:
0711 1110
0711 1111 R0 = NEXT CHARACTER IN INPUT BUFFER
0711 1112 R1/R2 = DESCRIPTOR OF UPDATED TOKEN
0711 1113
0711 1114 THE EXPANSION BUFFER WILL BE OVERWRITTEN WITH SYMBOL VALUE
0711 1115
0711 1116 ---
0711 1117 AMPERSAND:
0711 1118
0713 1119 PUSH R2,R3,R4,R5 ;SAVE REGISTERS
0715 1120 INCL R2 ;POINT TO SYMBOL NAME
0717 1121 DECL R1 ;ADJUST LENGTH OF SYMBOL NAME
0719 1122 BEQ 40$ ;BRANCH IF NULL SYMBOL NAME
071C 1123 BSBW DCL$SYN_STRING ;SEARCH LOCAL/GLOBAL SYMBOL TABLES
0721 1124 MOV R1,(R2),@ (SP) ;OVERWRITE '&SYMBOL' WITH ITS VALUE
0726 1125 POP R2,R3,R4,R5 ;SET NEW EXPANSION BUFFER POINTER
0728 1126 BSBW DCL$SETCHAR ;RESTORE REGISTERS
072B 1127 SUBL R2,WRK_L_EXPANDPTR(R10),R1 ;PEEK AT NEXT CHARACTER IN INPUT BUFFER
0731 1128 RSB ;CALCULATE LENGTH OF NEW TOKEN

00 BE 62 F8E4' 3C BB 0711 1118
F486 CA 51 D6 0713 1119
03 13 0715 1120
51 28 0717 1121
53 D0 0719 1122
3C BA 071C 1123
F8D5' 30 0721 1124
52 C3 0726 1125
05 0728 1126
072B 1127
0731 1128

40$:
PUSH R2,R3,R4,R5
INCL R2
DECL R1
BEQ 40$
BSBW DCL$SYN_STRING
MOV R1,(R2),@ (SP)
POP R2,R3,R4,R5
BSBW DCL$SETCHAR
SUBL R2,WRK_L_EXPANDPTR(R10),R1
RSB
```

```

0732 1130 .SBTTL PROCESS @FILESPEC CONSTRUCT
0732 1131 ---
0732 1132
0732 1133 HANDLE @FILESPEC CONSTRUCT.
0732 1134
0732 1135 INPUTS:
0732 1136
0732 1137 RO = CHARACTER IN INPUT BUFFER (@)
0732 1138
0732 1139 AT THIS POINT, ONLY THE @ HAS BEEN SEEN, AND THE FILESPEC
0732 1140 HAS NOT YET BEEN PROCESSED.
0732 1141
0732 1142 OUTPUTS:
0732 1143
0732 1144 RO = NEXT CHARACTER IN INPUT BUFFER
0732 1145 ---
0732 1146 INDIRECT:
19 68 AB 05 E2 0732 1147 BBSS #PRC_V_IND,PRC_W_FLAGS(R11),90$ ;SKIP IF INDIRECTION DISABLED
0737 1148 ;AND DISABLE WHILE PROCESSING FILESPEC
F48A CA DD 0737 1149 PUSHL WRK_L_MARKPTR(R10) ;SAVE OLD PARSE POSITION
F8C2' 30 0738 1150 BSBW DCL$MARK ;SET PARSE POSITION FOR '@' PROCESSING
F8BF' 30 073E 1151 BSBW DCL$STACKIND ;STACK CURRENT INDIRECT LEVEL
09 50 E9 0741 1152 CLRBIT PRC_V_IND,PRC_W_FLAGS(R11) ;ENABLE INDIRECTION AGAIN
F48A CA 8E D0 0745 1153 BLBC RO,DCL$CHARERROR ;BRANCH IF ERROR DETECTED
F8B5 30 0748 1154 MOVL (SP)+,WRK_L_MARKPTR(R10) ;RESTORE OLD PARSE POSITION
05 0740 1155 BSBW DCL$INPUT ;GET FIRST LINE, CHARACTER OF PROCEDURE
0750 1156 90$: RSB

```



```

0751 1158 .SBTTL ERROR HANDLER IN CHARACTER INPUT ROUTINES
0751 1159 :---
0751 1160 :
0751 1161 THIS ROUTINE IS CALLED TO PERFORM ANY SPECIAL PROCESSING WHEN AN
0751 1162 ERROR IS DETECTED BY THE CHARACTER INPUT ROUTINES.
0751 1163 :
0751 1164 INPUTS:
0751 1165 :
0751 1166 RO = STATUS CODE
0751 1167 :
0751 1168 OUTPUTS:
0751 1169 :
0751 1170 NONE
0751 1171 :---
0751 1172 DCL$CHARERROR::
F8AC' 30 0751 1173 BSBW DCL$ERRORMSG ;REPORT ERROR MESSAGE
SO DD 0754 1174 PUSHL RO ;SAVE STATUS CODE
F8A7' 30 0756 1175 BSBW DCL$FLUSH ;FLUSH INPUT BUFFER
SO BEDD 0759 1176 POPL RO ;RESTORE STATUS CODE
F8A1' 30 075C 1177 BSBW DCL$SET STATUS ;SET COMPLETION STATUS
F89E' 31 075F 1178 BRW DCL$RESTART ;START THE PARSING ALL OVER AGAIN

```

```
0762 1180 .SBTTL RECALL COMMAND
0762 1181
0762 1182 :+ DCL$RECALL - RECALL COMMAND
0762 1183
0762 1184 : THIS ROUTINE IS CALLED TO EXECUTE THE DCL RECALL COMMAND. THE RECALL
0762 1185 : COMMAND REPROMPTS THE USER WITH A COMMAND THAT HE HAS PREVIOUSLY ENTERED
0762 1186 : OR DISPLAYS FOR THE USER THE LIST OF ALL THE COMMANDS IN THE COMMAND BUFFER.
0762 1187
0762 1188 : INPUTS:
0762 1189 :
0762 1190 : R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR
0762 1191 : R9 = ADDRESS OF SCRATCH STACK
0762 1192 : R10 = ADDRESS OF COMMAND WORK AREA
0762 1193 : R11 = ADDRESS OF PROCESS WORK AREA
0762 1194
0762 1195 : OUTPUTS:
0762 1196 :
0762 1197 : R0 = STATUS CODE
0762 1198 :-
0762 1199
0762 1200 DCL$RECALL::
0762 1201
0762 1202 :
0762 1203 : SKIP IF ENTERED FROM A COMMAND PROCEDURE.
0762 1204 :
0762 1205 : BBS #PRC V MODE,- : IF SET, NOT INTERACTIVE
0762 1206 : PRC Q FLAGS(R11),3$ :
0762 1207 : BBS #PRC V YLEVEL,- : IF SET, AT CONTROL Y/C LEVEL
0762 1208 : PRC Q FLAGS(R11),5$ :
0762 1209 : TSTL PRC_L_INDEPTH(R11) : INDIRECT LEVEL ZERO?
0762 1210 : BEQL 5$ : BRANCH IF YES
0762 1211 3$: BRW 90$ : RETURN
0762 1212
0762 1213
0762 1214 :
0762 1215 : REMOVE THIS COMMAND FROM THE RECALL BUFFER.
0762 1216 :
0762 1217 5$: MOVL PRC_L_RECALLPTR(R11),- : UPDATE WRK RECALL PTR
0762 1218 : WRK_L_RECALLPTR(R10) :
0762 1219 : CLRQ -(SP) : ALLOCATE COMMAND DESCRIPTOR
0762 1220 : PUSHL SP : POINT TO IT
0762 1221 : CALLS #1,DCL$GET_PREV_COMMAND : GET THE RECALL COMMAND
0762 1222 : CLRQ (SP)+ : DISCARD THE DESCRIPTOR
0762 1223 : MOVL WRK_L_RECALLPTR(R10),- : UPDATE PRC RECALL PTR
0762 1224 : PRC_L_RECALLPTR(R11) :
0762 1225
0762 1226 :
0762 1227 : PARSE THE COMMAND.
0762 1228 :
0762 1229 : MOVL #1,R6 : ASSUME BACKING UP ONE COMMAND
0762 1230 : BSBW DCL$GETDVAL : GET NEXT DESCRIPTOR
0762 1231 : CMPL #PTR_K_ENDLINE,R5 : EOL?
0762 1232 : BEQL 30$ : YES, EXECUTE THE COMMAND
0762 1233 : CMPL #PTR_K_PARAMETR,R5 : NUMBER?
0762 1234 : BNEQ 20$ : NO, MUST BE /ALL
0762 1235 : MOVQ R1,R2 : COPY DESCRIPTOR
0762 1236 : MOVQ R2,-(SP) : SAVE IT ON THE STACK
```

0A 68 AB E0 0762 1205 BBS #PRC V MODE,- : IF SET, NOT INTERACTIVE  
0B 68 AB E0 0762 1206 PRC Q FLAGS(R11),3\$ :  
5C AB D5 0767 1207 BBS #PRC V YLEVEL,- : IF SET, AT CONTROL Y/C LEVEL  
03 13 0769 1208 PRC Q FLAGS(R11),5\$ :  
0113 31 076C 1209 TSTL PRC\_L\_INDEPTH(R11) : INDIRECT LEVEL ZERO?  
076F 1210 BEQL 5\$ : BRANCH IF YES  
0771 1211 3\$: BRW 90\$ : RETURN  
0774 1212  
0774 1213  
0774 1214  
0774 1215 : REMOVE THIS COMMAND FROM THE RECALL BUFFER.  
0774 1216 :  
012F CB D0 0774 1217 5\$: MOVL PRC\_L\_RECALLPTR(R11),- : UPDATE WRK RECALL PTR  
EA AA 0778 1218 : WRK\_L\_RECALLPTR(R10) :  
7E 7C 077A 1219 : CLRQ -(SP) : ALLOCATE COMMAND DESCRIPTOR  
5E DD 077C 1220 : PUSHL SP : POINT TO IT  
0000'CF 01 FB 077E 1221 : CALLS #1,DCL\$GET\_PREV\_COMMAND : GET THE RECALL COMMAND  
8E 7C 0783 1222 : CLRQ (SP)+ : DISCARD THE DESCRIPTOR  
EA AA D0 0785 1223 : MOVL WRK\_L\_RECALLPTR(R10),- : UPDATE PRC RECALL PTR  
012F CB 0788 1224 : PRC\_L\_RECALLPTR(R11) :  
0788 1225  
0788 1226 :  
0788 1227 : PARSE THE COMMAND.  
0788 1228 :  
56 01 D0 078B 1229 : MOVL #1,R6 : ASSUME BACKING UP ONE COMMAND  
F86F' 30 078E 1230 : BSBW DCL\$GETDVAL : GET NEXT DESCRIPTOR  
55 04 D1 0791 1231 : CMPL #PTR\_K\_ENDLINE,R5 : EOL?  
6C 13 0794 1232 : BEQL 30\$ : YES, EXECUTE THE COMMAND  
55 03 D1 0796 1233 : CMPL #PTR\_K\_PARAMETR,R5 : NUMBER?  
24 12 0799 1234 : BNEQ 20\$ : NO, MUST BE /ALL  
52 51 7D 079B 1235 : MOVQ R1,R2 : COPY DESCRIPTOR  
7E 52 7D 079E 1236 : MOVQ R2,-(SP) : SAVE IT ON THE STACK

```
51 01 D0 07A1 1237 MOVL #1,R1 : SET DECIMAL RADIX
    FB59 30 07A4 1238 BSBW DCL$CNVNOEDIT : CONVERT NUMBER TO BINARY
52 8E 7D 07A7 1239 MOVQ (SP)+,R2 : RESTORE THE DESCRIPTOR
    50 D5 07AA 1240 TSTL R0 : WAS IT A NUMBER?
    05 13 07AC 1241 BEQL 10$ : YES, CHECK BOUNDS
56 52 7D 07AE 1242 MOVQ R2,R6 : SAVE THE STRING DESCRIPTOR
    19 11 07B1 1243 BRB 35$ : GET THE COMMAND
14 51 D1 07B3 1244 10$: CMPL R1,#WRK_C_RECALLMAX : WITHIN BOUNDS?
    0C 1A 07B6 1245 BGTRU 95$ : NO, SIGNAL ERROR
56 51 D0 07B8 1246 MOVL R1,R6 : GET BACKUP COUNT
    07 13 07BB 1247 BEQL 95$ : SIGNAL ERROR IF ZERO
    43 11 07BD 1248 BRB 30$ : GET THE COMMAND
56 14 D0 07BF 1249 20$: MOVL #WRK_C_RECALLMAX,R6 : /ALL WAS SPECIFIED
    5F 11 07C2 1250 BRB 50$ : DISPLAY THE COMMANDS
    07C4 1251
    07C4 1252 95$: STATUS IVVALU : SET INVALID VALUE STATUS
    05 07CB 1253 RSB : RETURN
    07CC 1254
    07CC 1255 :
    07CC 1256 : FETCH SPECIFIED COMMAND BY STRING.
    07CC 1257
59 14 D0 07CC 1258 35$: MOVL #WRK_C_RECALLMAX,R9 : /ALL WAS SPECIFIED
    7E 7C 07CF 1259 CLRQ -(SP) : ALLOCATE COMMAND DESCRIPTOR
    5E DD 07D1 1260 36$: PUSHL SP : PUSH DESCR ADDRESS
    04 B8 0000'CF 01 FB 07D3 1261 CALLS #1,DCL$GET_PREV_COMMAND : GET THE PREVIOUS COMMAND
    04 BE 6E 28 07D8 1262 MOVQ3 (SP),R4(R8) : COPY COMMAND TO SCRATCH BUFFER
    68 6E D0 07DE 1263 MOVL (SP),R8 : COPY INITIAL LENGTH
    51 68 7D 07E1 1264 MOVQ (R8),R1 : SET COMMAND DESCRIPTOR
    FB19 30 07E4 1265 BSBW DCL$TRIM : UPCASE AND TRIM THE COMMAND
    51 56 D1 07E7 1266 CMPL R6,R1 : MAKE SURE WE ARE CHECKING A SUBSTRING
    08 1A 07EA 1267 BGTRU 361$ : SKIP THIS ONE IF NOT
62 56 00 67 56 2D 07EC 1268 CMPC5 R6,(R7),#0,R6,(R2) : DO THE STRINGS MATCH?
    1A 13 07F2 1269 BEQL 37$ : YES, THEN REPROMPT
    DA 59 F5 07F4 1270 361$: SOBGTR R9,36$ : LOOP TILL SEARCHED ALL COMMANDS
    SE 08 C0 07F7 1271 ADDL #8,SP : RESTORE THE STACK
50 00038238 8F D0 07FA 1272 MOVL #CLIS_CMDNOTFND,R0 : SET COMMAND NOT FOUND STATUS
    05 0801 1273 RSB : RETURN
    0802 1274
    0802 1275 :
    0802 1276 : FETCH SPECIFIED COMMAND BY NUMBER.
    0802 1277
    7E 7C 0802 1278 30$: CLRQ -(SP) : ALLOCATE COMMAND DESCRIPTOR
    5E DD 0804 1279 32$: PUSHL SP : PUSH DESCR ADDRESS
    0000'CF 01 FB 0806 1280 CALLS #1,DCL$GET_PREV_COMMAND : GET THE PREVIOUS COMMAND
    F6 56 F5 080B 1281 SOBGTR R6,32$ : LOOP TILL BACKED UP FAR ENOUGH
    080E 1282
    080E 1283 :
    080E 1284 : SET UP RAB TO REPROMPT WITH THE COMMAND.
    080E 1285
    51 8E 7D 080E 1286 37$: MOVQ (SP)+,R1 : GET COMMAND DESCRIPTOR
54 14 AB D0 0811 1287 MOVL PRC_L_INDINPRAB(R11),R4 : ASSUME USING INDIRECT INPUT RAB
    0B E1 0815 1288 BBC #PRC_Y_LEVEL - : BRANCH IF NOT AT CTRL/Y LEVEL
    04 68 AB 0817 1289 PRC_Q_FLAGS(R11),40$
54 08 AB D0 081A 1290 MOVL PRC_L_INPRAB(R11),R4 : USE LEVEL 0 INPUT RAB
    FBCE 30 081E 1291 40$: BSBW INSERT_COMMAND : INSERT THE COMMAND
    64 11 0821 1292 BRB 90$
    0823 1293
```



```
0823 1294 :  
0823 1295 : DISPLAY ALL COMMANDS  
0823 1296 :  
20312000 8F D0 0823 1297 50$: MOVL #^X20312000,- : SET INITIAL NUMBER STRING  
F890 CA 0823 1298 WRK_G_INPBUF-6(R10) :  
50 EA AA D0 0823 1299 55$: MOVL WRK_L_RECALLPTR(R10),R0 : HAVE WE ALREADY DISPLAYED THEM ALL?  
50 FF AO 9E 0830 1300 MOVAB -1(R0),R0 :  
51 0133 CB 9E 0834 1301 MOVAB PRC_G_COMMANDS(R11),R1 : WRAP IF NECESSARY  
51 50 D1 0839 1302 CMPL R0,R1 :  
05 1E 083C 1303 BGEQU 57$ :  
50 0533 CB 9E 083E 1304 57$: MOVL PRC_G_COMMANDS+PRC_C_CMDBUFSIZ-1(R11),R0 :  
60 95 0843 1305 TSTB (R0) : BRANCH IF NO COMMANDS LEFT  
40 13 0845 1306 BEQL 90$ :  
7E 7C 0847 1307 CLRQ -(SP) :  
5E DD 0849 1308 PUSHL SP : ALLOCATE COMMAND DESCRIPTOR  
0000 CF 01 FB 084B 1309 CALLS #1,DCL$GET_PREV_COMMAND : PUSH DESCR ADDRESS  
51 8E 7D 0850 1310 MOVQ (SP)+,R1 : GET THE PREVIOUS COMMAND  
51 03 C0 0853 1311 ADDL #3,R1 : GET COMMAND DESCRIPTOR  
52 03 C2 0856 1312 SUBL #3,R2 : INSERT THE COMMAND NUMBER  
7E 51 7D 0859 1313 MOVQ R1,-(SP) :  
F7A1 30 085C 1314 BSBW DCL$MSGOUT : SAVE COMMAND DESCRIPTOR  
51 8E 7D 085F 1315 MOVQ (SP)+,R1 : OUTPUT THE COMMAND  
39 01 A2 91 0862 1316 CMPB 1(R2),#^X39 : RESTORE COMMAND DESCRIPTOR  
11 12 0866 1317 BNEQ 60$ : IS ONES DIGIT A 9?  
31 62 91 0868 1318 CMPB (R2),#^X31 : NO, THEN SKIP  
07 12 086B 1319 BNEQ 58$ : IS TENS DIGIT A 1?  
62 2F32 8F B0 086D 1320 MOVW #^X2F32,(R2) : NO, THEN SKIP  
05 11 0872 1321 BRB 60$ : YES, INSERT '2/'  
62 2F31 8F B0 0874 1322 58$: MOVW #^X2F31,(R2) : BRANCH  
01 A2 96 0879 1323 60$: INCB 1(R2) : INSERT '1/'  
087C 1324 : INCREMENT THE COMMAND NUMBER  
087C 1325 :  
087C 1326 : CHECK FOR NO MORE COMMANDS.  
087C 1327 :  
012F CB D1 087C 1328 CMPL PRC_L_RECALLPTR(R11),- : HAVE WE RETURNED TO THE ORIGIN?  
EA AA 0880 1329 WRK_L_RECALLPTR(R10) :  
03 13 0882 1330 BEQL 90$ :  
A5 56 F5 0884 1331 SOBGTR R6,55$ : OR DISPLAYED THE MAX # OF COMMANDS?  
0887 1332 :  
0887 1333 90$: STATUS NORMAL : SET SUCCESS  
088E 1334 RSB : ALL DONE  
088F 1335 :  
088F 1336 .END
```

READREC  
Symbol table

- READ AN INPUT RECORD

B 10

16-SEP-1984 00:11:48 VAX/VMS Macro V04-00  
4-SEP-1984 23:42:34 [DCL.SRC]READREC.MAR;1

Page 33  
(18)

|                       |   |          |    |    |
|-----------------------|---|----------|----|----|
| SS.TMP1               | = | 00000001 |    |    |
| SS.TMP2               | = | 00000065 |    |    |
| SS1                   | = | 00000001 |    |    |
| ABORT                 |   | 0000028D | R  | 02 |
| AMPERSAND             |   | 00000711 | R  | 02 |
| CLIS_BUFOVF           | = | 00038018 |    |    |
| CLIS_CMDNOTFND        | = | 00038238 |    |    |
| CLIS_EXPSYN           | = | 00038038 |    |    |
| CLIS_IVVALU           | = | 00038088 |    |    |
| CLIS_NORMAL           | = | 00030001 |    |    |
| CLIS_SYMOVF           | = | 00038138 |    |    |
| CLIS_USGOTO           | = | 00038148 |    |    |
| DCL\$ABORT            |   | *****    | X  | 02 |
| DCL\$ALLOC_STATE      |   | *****    | X  | 02 |
| DCL\$BACKUP_CHAR      |   | *****    | X  | 02 |
| DCL\$CHARERROR        |   | 00000751 | RG | 02 |
| DCL\$CLOSE_PPFS       |   | *****    | X  | 02 |
| DCL\$CHVNOEDIT        |   | *****    | X  | 02 |
| DCL\$CRLF             |   | *****    | X  | 02 |
| DCL\$CVT_STRING       |   | *****    | X  | 02 |
| DCL\$DEACGOTO         |   | *****    | X  | 02 |
| DCL\$DEALLOC_STATE    |   | *****    | X  | 02 |
| DCL\$DISABLE          |   | *****    | X  | 02 |
| DCL\$DSBCONTRLY       |   | *****    | X  | 02 |
| DCL\$ERRORMSG         |   | *****    | X  | 02 |
| DCL\$FLUSH            |   | *****    | X  | 02 |
| DCL\$GETCHAR          |   | *****    | X  | 02 |
| DCL\$GETDVAL          |   | *****    | X  | 02 |
| DCL\$GETOKEN          |   | *****    | X  | 02 |
| DCL\$GET_CURR_COMMAND |   | *****    | X  | 02 |
| DCL\$GET_NEXT_COMMAND |   | *****    | X  | 02 |
| DCL\$GET_PREV_COMMAND |   | *****    | X  | 02 |
| DCL\$INPUT            |   | 00000005 | RG | 02 |
| DCL\$LEXIF            |   | *****    | X  | 02 |
| DCL\$LOCKED_STATE     |   | 000005B1 | RG | 02 |
| DCL\$MARK             |   | *****    | X  | 02 |
| DCL\$MSGOUT           |   | *****    | X  | 02 |
| DCL\$PARSERR          |   | *****    | X  | 02 |
| DCL\$PUTCHAR          |   | *****    | X  | 02 |
| DCL\$PUT_COMMAND      |   | *****    | X  | 02 |
| DCL\$PUT_SEGMENT      |   | *****    | X  | 02 |
| DCL\$RECALL           |   | 00000762 | RG | 02 |
| DCL\$RESTART          |   | *****    | X  | 02 |
| DCL\$SEARCH           |   | *****    | X  | 02 |
| DCL\$SEARCH_KEYPAD    |   | *****    | X  | 02 |
| DCL\$SETCHAR          |   | *****    | X  | 02 |
| DCL\$SET_STATUS       |   | *****    | X  | 02 |
| DCL\$SPECIAL          |   | 00000705 | RG | 02 |
| DCL\$STACKIND         |   | *****    | X  | 02 |
| DCL\$SYM_STRING       |   | *****    | X  | 02 |
| DCL\$TRIM             |   | *****    | X  | 02 |
| DCL\$UNSTACK          |   | *****    | X  | 02 |
| DCL\$UPCASE           |   | *****    | X  | 02 |
| DEV\$V_TRM            |   | *****    | X  | 02 |
| DVIS_DEVDEPEND2       | = | 0000001C |    |    |
| END_OF_LIST           |   | 00000322 | R  | 02 |
| ERASE                 |   | 00000000 | R  | 02 |

|                  |          |          |    |
|------------------|----------|----------|----|
| ERASE_LINE       | 0000037F | R        | 02 |
| ERROR            | 000000AE | R        | 02 |
| EXESC_SYSEFN     | *****    | X        | 02 |
| EXPAND           | 000005CA | R        | 02 |
| FABSM_CR         | =        | 00000002 |    |
| FABSM_IFI        | =        | 00000002 |    |
| GET_INPUT        | 00000045 | R        | 02 |
| GET_KEY_NAME     | *****    | X        | 02 |
| INDIRECT         | 00000732 | R        | 02 |
| INSERT_COMMAND   | 000003EF | R        | 02 |
| IO_ERROR         | 0000019B | R        | 02 |
| ITRM_C_LENGTH    | 00000030 |          |    |
| ITRM_C_MINLEN    | 00000018 |          |    |
| ITRM_K_LENGTH    | 00000030 |          |    |
| ITRM_K_MINLEN    | 00000018 |          |    |
| ITRM_L_INIADDR   | 0000001C |          |    |
| ITRM_L_INIRET    | 00000020 |          |    |
| ITRM_L_MODIFIERS | 00000004 |          |    |
| ITRM_L_MODRET    | 00000008 |          |    |
| ITRM_L_OFFRET    | 0000002C |          |    |
| ITRM_L_OFFSET    | 00000028 |          |    |
| ITRM_L_PMPTADDR  | 00000010 |          |    |
| ITRM_L_PMPTRET   | 00000014 |          |    |
| ITRM_W_INICODE   | 0000001A |          |    |
| ITRM_W_INILEN    | 00000018 |          |    |
| ITRM_W_MODCODE   | 00000002 |          |    |
| ITRM_W_MODLEN    | 00000000 |          |    |
| ITRM_W_OFFCODE   | 00000026 |          |    |
| ITRM_W_OFFLEN    | 00000024 |          |    |
| ITRM_W_PMPTCODE  | 0000000E |          |    |
| ITRM_W_PMPTLEN   | 0000000C |          |    |
| PRC_B_CONTINUE   | 000000F3 |          |    |
| PRC_B_DEFRADIX   | 000000AE |          |    |
| PRC_B_EXMDEPMOD  | 000000AD |          |    |
| PRC_B_EXMDEPWID  | 000000AC |          |    |
| PRC_B_EXONLYL    | 0000012D |          |    |
| PRC_B_FLAGS2     | 000000AF |          |    |
| PRC_B_IMGFLAG    | 00000078 |          |    |
| PRC_B_OUTFLAGS   | 0000012C |          |    |
| PRC_B_PROMPTLEN  | 000000F0 |          |    |
| PRC_C_CMDBUFSIZ  | =        | 00000401 |    |
| PRC_C_LENGTH     | 00000534 |          |    |
| PRC_G_COMMANDS   | 00000133 |          |    |
| PRC_G_PROMPT     | 000000F4 |          |    |
| PRC_K_LENGTH     | 00000534 |          |    |
| PRC_L_CURRKEY    | 00000048 |          |    |
| PRC_L_EXMDEPADR  | 000000A8 |          |    |
| PRC_L_EXTARG     | 00000094 |          |    |
| PRC_L_EXTBLK     | 0000008C |          |    |
| PRC_L_EXTCOD     | 0000009C |          |    |
| PRC_L_EXTHND     | 00000090 |          |    |
| PRC_L_EXTPRM     | 00000098 |          |    |
| PRC_L_IDFLNK     | 0000008C |          |    |
| PRC_L_IMGACTSTS  | 00000080 |          |    |
| PRC_L_INDCLOCK   | 0000007C |          |    |
| PRC_L_INDEPTH    | 0000005C |          |    |
| PRC_L_INDFAB     | 0000001C |          |    |

READREC  
Symbol table

- READ AN INPUT RECORD

C 10

16-SEP-1984 00:11:48 VAX/VMS Macro V04-00  
4-SEP-1984 23:42:34 [DCL.SRC]READREC.MAR;1

Page 34  
(18)

```

PRC_L_INDINPRAB      00000014
PRC_L_INDOUTRAB      00000018
PRC_L_INPRAB         00000008
PRC_L_LASTKEY        0000004C
PRC_L_LSTSTATUS      000000B0
PRC_L_ONCTLY         000000B8
PRC_L_ONERROR        0000006C
PRC_L_OUTOFBAND      000000B4
PRC_L_OUTRAB         0000000C
PRC_L_OUTRABCTX      00000118
PRC_L_PPFLIST        00000070
PRC_L_RECALLPTR      0000012F
PRC_L_RESTART        00000058
PRC_L_SAVAP          00000000
PRC_L_SAVFP          00000004
PRC_L_SEVERITY       00000050
PRC_L_SPWN           000000C0
PRC_L_STACKLM        000000A4
PRC_L_STACKPT        000000A0
PRC_L_STATUS         00000054
PRC_L_STS            00000084
PRC_L_STV            00000088
PRC_L_SYMBOL         00000060
PRC_L_TMBX           00000074
PRC_L_TRMLIST        00000010
PRC_Q_ALLOCREG       00000020
PRC_Q_COMMAND        000000E0
PRC_Q_FLUSHTIME      000000D0
PRC_Q_GLOBAL         00000028
PRC_Q_IMAGENAME      000000D8
PRC_Q_KEYPAD         00000040
PRC_Q_LABEL          00000030
PRC_Q_LOCAL          00000038
PRC_Q_SAVEPRIV       000000E8
PRC_T_OUTDVI         0000C11C
PRC_V_AUTOLOGO       = 00000008
PRC_V_CARRCNTL       = 00000000
PRC_V_CNTRL          = 00000001
PRC_V_EOFLOGO        = 0000000E
PRC_V_FLUSH          = 00000006
PRC_V_GOTO           = 00000004
PRC_V_IND            = 00000005
PRC_V_MODE           = 00000006
PRC_V_VERIFY         = 00000007
PRC_V_YLEVEL         = 0000000B
PRC_W_ASTIOSB        000000C6
PRC_W_ASTRETN        000000C8
PRC_W_ASTSTATUS      000000C4
PRC_W_ATTMBX         0000007A
PRC_W_FLAGS          00000068
PRC_W_INPCHAN        00000064
PRC_W_ONLEVEL        0000006A
PRC_W_OUTIFI         00000114
PRC_W_OUTISI         00000116
PRC_W_OUTMBXCHN      000000CA
PRC_W_OUTMBXREF      000000CE
PRC_W_OUTMBXSIZ      000000CC

```

```

PRC_W_PMPTCTRL       000000F1
PRC_W_WAITIOSB       00000066
PROCESS_ESCAPE       00000407 R R 02
PROCESS_INPUT        000000B1 R R 02
PROCESS_RECALL       000002C5 R 02
PTR_B_LEVEL          00000004
PTR_B_NUMBER         00000005
PTR_B_PARMCNT        00000006
PTR_B_VALUE          00000000
PTR_C_LENGTH         = 00000004
PTR_K_ENDLINE        = 0000000C
PTR_K_LENGTH         = 00000003
PTR_K_PARAMETR       00000000
PTR_L_DESCR          00000008
PTR_L_ENTITY         = 00000018
RAB$L_CTX            = 00000028
RAB$L_RBF            = 00000008
RAB$L_STS            = 0000000C
RAB$L_STV            = 00000024
RAB$L_UBF            = 00000040
RAB$L_XAB            = 00000008
RAB$S_PPF_RAT        = 0000000E
RAB$V_PPF_IND        = 00000006
RAB$V_PPF_RAT        = 00000002
RAB$W_ISI            = 00000022
RAB$W_RSZ            = 0000000C
RAB$W_STV0           = 0000000E
RAB$W_STV2           = 00000020
RAB$W_USZ            = 00000315 R R 02
RECALL_CURR          0000032E R R 02
RECALL_NEXT          000002CF R R 02
RECALL_PREV          00000007 R R 02
REINP                0000017F R 02
RETURN               ***** X 02
RMSS_CONTROLY        ***** X 02
RMSS_EOF              ***** X 02
RMSS_RSA              ***** X 02
RMSS_SYS              ***** X 02
SILENT_LOGOUT        0000029B RG 02
SPECIAL              00000196 R 02
SS$EXQUOTA           ***** X 02
STATUS               00000287 R 02
SYM_B_FLAGS          0000000B
SYM_B_NONUNIQUE      0000000B
SYM_B_TYPE           0000000A
SYM_L_BL             00000004
SYM_L_FL             00000000
SYM_T_SYMBOL         0000000C
SYM_V_ECHO           = 00000000
SYM_V_ERASE          = 00000004
SYM_V_LOCK           = 00000003
SYM_V_STATE          = 00000002
SYM_V_TERMINATE      = 00000001
SYM_W_SIZE           00000008
SYSS$CANCEL          ***** GX 02
SYSS$CANEXH          ***** GX 02
SYSS$CLOSE           ***** GX 02

```



READREC  
Symbol table

- READ AN INPUT RECORD

D 10

16-SEP-1984 00:11:48 VAX/VMS Macro V04-00  
4-SEP-1984 23:42:34 [DCL.SRC]READREC.MAR;1

Page 35  
(18)

|                 |            |    |    |                   |            |
|-----------------|------------|----|----|-------------------|------------|
| SYS\$EXIT       | *****      | GX | 02 | WRK_W_PMPTLEN     | FFFFF99E   |
| SYS\$GET        | *****      | GX | 02 | XAB\$W_ITMLST_LEN | = 0000000C |
| SYS\$GETDVIW    | *****      | GX | 02 | SS_               | = 000000EF |
| SYS\$PUT        | *****      | GX | 02 |                   |            |
| SYS\$WAIT       | *****      | GX | 02 |                   |            |
| TT2\$V_ANSICRT  | = 00000018 |    |    |                   |            |
| WRK_B_CMDOPT    | FFFFFFC3   |    |    |                   |            |
| WRK_B_MAXPARM   | FFFFFFD0   |    |    |                   |            |
| WRK_B_MINPARM   | FFFFFFD1   |    |    |                   |            |
| WRK_B_PARMCNT   | FFFFFFCE   |    |    |                   |            |
| WRK_B_PARMSUM   | FFFFFFCF   |    |    |                   |            |
| WRK_B_RECALLCNT | FFFFFFC5   |    |    |                   |            |
| WRK_B_VALLEV    | FFFFFFC4   |    |    |                   |            |
| WRK_B_VERBTYP   | FFFFFFC2   |    |    |                   |            |
| WRK_C_INPBUFSIZ | = 00000100 |    |    |                   |            |
| WRK_C_LENGTH    | FFFFFF486  |    |    |                   |            |
| WRK_C_RECALLMAX | = 00000014 |    |    |                   |            |
| WRK_G_BUFFER    | FFFFFF492  |    |    |                   |            |
| WRK_G_INPBUF    | FFFFFF896  |    |    |                   |            |
| WRK_G_RESULT    | FFFFFF9B6  |    |    |                   |            |
| WRK_K_LENGTH    | FFFFFF486  |    |    |                   |            |
| WRK_L_CHARPTR   | FFFFFF48E  |    |    |                   |            |
| WRK_L_DISALLOW  | FFFFFFE6   |    |    |                   |            |
| WRK_L_ERRORRTN  | FFFFFF9AE  |    |    |                   |            |
| WRK_L_EXPANDPTR | FFFFFF486  |    |    |                   |            |
| WRK_L_IMAGE     | FFFFFFE2   |    |    |                   |            |
| WRK_L_MARKPTR   | FFFFFF48A  |    |    |                   |            |
| WRK_L_PAROUT    | FFFFFFD2   |    |    |                   |            |
| WRK_L_PMPTADDR  | FFFFFF9A2  |    |    |                   |            |
| WRK_L_PROMPTRTN | FFFFFF9A6  |    |    |                   |            |
| WRK_L_PROPTR    | FFFFFFFC6  |    |    |                   |            |
| WRK_L_QUABLK    | FFFFFFCA   |    |    |                   |            |
| WRK_L_READRTN   | FFFFFF9AA  |    |    |                   |            |
| WRK_L_RECALLPTR | FFFFFFFEA  |    |    |                   |            |
| WRK_L_RSLEND    | FFFFFFB6   |    |    |                   |            |
| WRK_L_RSLNXT    | FFFFFFBA   |    |    |                   |            |
| WRK_L_SAVAP     | FFFFFFF8   |    |    |                   |            |
| WRK_L_SAVFP     | FFFFFFFC   |    |    |                   |            |
| WRK_L_SAVSP     | FFFFFFF4   |    |    |                   |            |
| WRK_L_SIGNALRTN | FFFFFFD6   |    |    |                   |            |
| WRK_L_SPECRTN   | FFFFFF9B2  |    |    |                   |            |
| WRK_L_TAB_VEC   | FFFFFFDE   |    |    |                   |            |
| WRK_L_VERB      | FFFFFFBE   |    |    |                   |            |
| WRK_M_INPSUBST  | = 00000400 |    |    |                   |            |
| WRK_M_NOUPCASE  | = 00000800 |    |    |                   |            |
| WRK_M_QUOTE     | = 00000010 |    |    |                   |            |
| WRK_M_STAR      | = 00000020 |    |    |                   |            |
| WRK_V_COMMAND   | = 00000001 |    |    |                   |            |
| WRK_V_COMMENT   | = 0000000C |    |    |                   |            |
| WRK_V_CONTIN    | = 00000003 |    |    |                   |            |
| WRK_V_INPSUBST  | = 0000000A |    |    |                   |            |
| WRK_V_INQUIRE   | = 00000007 |    |    |                   |            |
| WRK_V_QUOTE     | = 00000004 |    |    |                   |            |
| WRK_V_TRAILSPC  | = 00000009 |    |    |                   |            |
| WRK_W_FLAGS     | FFFFFFF0   |    |    |                   |            |
| WRK_W_FLAGS2    | FFFFFFF2   |    |    |                   |            |
| WRK_W_IMGCHAN   | FFFFFFEE   |    |    |                   |            |



-----  
! Psect synopsis !  
-----

| PSECT name | Allocation        | PSECT No. | Attributes |     |     |     |     |       |       |      |       |       |      |  |  |  |  |  |  |
|------------|-------------------|-----------|------------|-----|-----|-----|-----|-------|-------|------|-------|-------|------|--|--|--|--|--|--|
| ABS        | 00000000 ( 0.)    | 00 ( 0.)  | NOPIC      | USR | CON | ABS | LCL | NOSHR | NOEXE | NORD | NOWRT | NOVEC | BYTE |  |  |  |  |  |  |
| \$ABSS     | FFFFFFFC ( 0.)    | 01 ( 1.)  | NOPIC      | USR | CON | ABS | LCL | NOSHR | EXE   | RD   | WRT   | NOVEC | BYTE |  |  |  |  |  |  |
| DCL\$ZCODE | 0000088F ( 2191.) | 02 ( 2.)  | NOPIC      | USR | CON | REL | LCL | NOSHR | EXE   | RD   | NOWRT | NOVEC | BYTE |  |  |  |  |  |  |

-----  
! Performance indicators !  
-----

| Phase                  | Page faults | CPU Time    | Elapsed Time |
|------------------------|-------------|-------------|--------------|
| Initialization         | 9           | 00:00:00.07 | 00:00:00.80  |
| Command processing     | 80          | 00:00:00.70 | 00:00:06.75  |
| Pass 1                 | 351         | 00:00:15.31 | 00:00:43.74  |
| Symbol table sort      | 0           | 00:00:01.60 | 00:00:04.80  |
| Pass 2                 | 235         | 00:00:03.98 | 00:00:10.02  |
| Symbol table output    | 33          | 00:00:00.23 | 00:00:00.64  |
| Psect synopsis output  | 2           | 00:00:00.03 | 00:00:00.26  |
| Cross-reference output | 0           | 00:00:00.00 | 00:00:00.00  |
| Assembler run totals   | 710         | 00:00:21.94 | 00:01:07.02  |

The working set limit was 1500 pages.  
79799 bytes (156 pages) of virtual memory were used to buffer the intermediate code.  
There were 60 pages of symbol table space allocated to hold 1072 non-local and 94 local symbols.  
1336 source lines were read in Pass 1, producing 22 object records in Pass 2.  
58 pages of virtual memory were used to define 41 macros.

-----  
! Macro library statistics !  
-----

| Macro library name                     | Macros defined |
|--|----------------|
| _\$255\$DUA28:[SYSLIB]SYSB LDMLB.MLB;1 | 0              |
| _\$255\$DUA28:[DCL.OBJ]DCL.MLB;1       | 13             |
| _\$255\$DUA28:[SYS.OBJ]LIB.MLB;1       | 0              |
| _\$255\$DUA28:[SYSLIB]STARLET.MLB;2    | 20             |
| TOTALS (all libraries)                 | 33             |

1301 GETS were required to define 33 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:READREC/OBJ=OBJ\$:READREC MSRC\$:READREC/UPDATE=(ENH\$:READREC)+EXECMLS/LIB+LIB\$:DCL/LIB+SYSS\$LIBRARY:SYSB LDMLB/LIB



0072 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

|     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 001 | 002 | 003 | 004 | 005 | 006 | 007 | 008 | 009 | 010 | 011 | 012 | 013 | 014 | 015 | 016 | 017 | 018 | 019 | 020 | 021 | 022 | 023 | 024 | 025 | 026 | 027 | 028 | 029 | 030 | 031 | 032 | 033 | 034 | 035 | 036 | 037 | 038 | 039 | 040 | 041 | 042 | 043 | 044 | 045 | 046 | 047 | 048 | 049 | 050 | 051 | 052 | 053 | 054 | 055 | 056 | 057 | 058 | 059 | 060 | 061 | 062 | 063 | 064 | 065 | 066 | 067 | 068 | 069 | 070 | 071 | 072 | 073 | 074 | 075 | 076 | 077 | 078 | 079 | 080 | 081 | 082 | 083 | 084 | 085 | 086 | 087 | 088 | 089 | 090 | 091 | 092 | 093 | 094 | 095 | 096 | 097 | 098 | 099 | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 | 128 | 129 | 130 | 131 | 132 | 133 | 134 | 135 | 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 | 144 | 145 | 146 | 147 | 148 | 149 | 150 | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 | 160 | 161 | 162 | 163 | 164 | 165 | 166 | 167 | 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 | 176 | 177 | 178 | 179 | 180 | 181 | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 | 192 | 193 | 194 | 195 | 196 | 197 | 198 | 199 | 200 | 201 | 202 | 203 | 204 | 205 | 206 | 207 | 208 | 209 | 210 | 211 | 212 | 213 | 214 | 215 | 216 | 217 | 218 | 219 | 220 | 221 | 222 | 223 | 224 | 225 | 226 | 227 | 228 | 229 | 230 | 231 | 232 | 233 | 234 | 235 | 236 | 237 | 238 | 239 | 240 | 241 | 242 | 243 | 244 | 245 | 246 | 247 | 248 | 249 | 250 | 251 | 252 | 253 | 254 | 255 | 256 | 257 | 258 | 259 | 260 | 261 | 262 | 263 | 264 | 265 | 266 | 267 | 268 | 269 | 270 | 271 | 272 | 273 | 274 | 275 | 276 | 277 | 278 | 279 | 280 | 281 | 282 | 283 | 284 | 285 | 286 | 287 | 288 | 289 | 290 | 291 | 292 | 293 | 294 | 295 | 296 | 297 | 298 | 299 | 300 | 301 | 302 | 303 | 304 | 305 | 306 | 307 | 308 | 309 | 310 | 311 | 312 | 313 | 314 | 315 | 316 | 317 | 318 | 319 | 320 | 321 | 322 | 323 | 324 | 325 | 326 | 327 | 328 | 329 | 330 | 331 | 332 | 333 | 334 | 335 | 336 | 337 | 338 | 339 | 340 | 341 | 342 | 343 | 344 | 345 | 346 | 347 | 348 | 349 | 350 | 351 | 352 | 353 | 354 | 355 | 356 | 357 | 358 | 359 | 360 | 361 | 362 | 363 | 364 | 365 | 366 | 367 | 368 | 369 | 370 | 371 | 372 | 373 | 374 | 375 | 376 | 377 | 378 | 379 | 380 | 381 | 382 | 383 | 384 | 385 | 386 | 387 | 388 | 389 | 390 | 391 | 392 | 393 | 394 | 395 | 396 | 397 | 398 | 399 | 400 | 401 | 402 | 403 | 404 | 405 | 406 | 407 | 408 | 409 | 410 | 411 | 412 | 413 | 414 | 415 | 416 | 417 | 418 | 419 | 420 | 421 | 422 | 423 | 424 | 425 | 426 | 427 | 428 | 429 | 430 | 431 | 432 | 433 | 434 | 435 | 436 | 437 | 438 | 439 | 440 | 441 | 442 | 443 | 444 | 445 | 446 | 447 | 448 | 449 | 450 | 451 | 452 | 453 | 454 | 455 | 456 | 457 | 458 | 459 | 460 | 461 | 462 | 463 | 464 | 465 | 466 | 467 | 468 | 469 | 470 | 471 | 472 | 473 | 474 | 475 | 476 | 477 | 478 | 479 | 480 | 481 | 482 | 483 | 484 | 485 | 486 | 487 | 488 | 489 | 490 | 491 | 492 | 493 | 494 | 495 | 496 | 497 | 498 | 499 | 500 | 501 | 502 | 503 | 504 | 505 | 506 | 507 | 508 | 509 | 510 | 511 | 512 | 513 | 514 | 515 | 516 | 517 | 518 | 519 | 520 | 521 | 522 | 523 | 524 | 525 | 526 | 527 | 528 | 529 | 530 | 531 | 532 | 533 | 534 | 535 | 536 | 537 | 538 | 539 | 540 | 541 | 542 | 543 | 544 | 545 | 546 | 547 | 548 | 549 | 550 | 551 | 552 | 553 | 554 | 555 | 556 | 557 | 558 | 559 | 560 | 561 | 562 | 563 | 564 | 565 | 566 | 567 | 568 | 569 | 570 | 571 | 572 | 573 | 574 | 575 | 576 | 577 | 578 | 579 | 580 | 581 | 582 | 583 | 584 | 585 | 586 | 587 | 588 | 589 | 590 | 591 | 592 | 593 | 594 | 595 | 596 | 597 | 598 | 599 | 600 | 601 | 602 | 603 | 604 | 605 | 606 | 607 | 608 | 609 | 610 | 611 | 612 | 613 | 614 | 615 | 616 | 617 | 618 | 619 | 620 | 621 | 622 | 623 | 624 | 625 | 626 | 627 | 628 | 629 | 630 | 631 | 632 | 633 | 634 | 635 | 636 | 637 | 638 | 639 | 640 | 641 | 642 | 643 | 644 | 645 | 646 | 647 | 648 | 649 | 650 | 651 | 652 | 653 | 654 | 655 | 656 | 657 | 658 | 659 | 660 | 661 | 662 | 663 | 664 | 665 | 666 | 667 | 668 | 669 | 670 | 671 | 672 | 673 | 674 | 675 | 676 | 677 | 678 | 679 | 680 | 681 | 682 | 683 | 684 | 685 | 686 | 687 | 688 | 689 | 690 | 691 | 692 | 693 | 694 | 695 | 696 | 697 | 698 | 699 | 700 | 701 | 702 | 703 | 704 | 705 | 706 | 707 | 708 | 709 | 710 | 711 | 712 | 713 | 714 | 715 | 716 | 717 | 718 | 719 | 720 | 721 | 722 | 723 | 724 | 725 | 726 | 727 | 728 | 729 | 730 | 731 | 732 | 733 | 734 | 735 | 736 | 737 | 738 | 739 | 740 | 741 | 742 | 743 | 744 | 745 | 746 | 747 | 748 | 749 | 750 | 751 | 752 | 753 | 754 | 755 | 756 | 757 | 758 | 759 | 760 | 761 | 762 | 763 | 764 | 765 | 766 | 767 | 768 | 769 | 770 | 771 | 772 | 773 | 774 | 775 | 776 | 777 | 778 | 779 | 780 | 781 | 782 | 783 | 784 | 785 | 786 | 787 | 788 | 789 | 790 | 791 | 792 | 793 | 794 | 795 | 796 | 797 | 798 | 799 | 800 | 801 | 802 | 803 | 804 | 805 | 806 | 807 | 808 | 809 | 810 | 811 | 812 | 813 | 814 | 815 | 816 | 817 | 818 | 819 | 820 | 821 | 822 | 823 | 824 | 825 | 826 | 827 | 828 | 829 | 830 | 831 | 832 | 833 | 834 | 835 | 836 | 837 | 838 | 839 | 840 | 841 | 842 | 843 | 844 | 845 | 846 | 847 | 848 | 849 | 850 | 851 | 852 | 853 | 854 | 855 | 856 | 857 | 858 | 859 | 860 | 861 | 862 | 863 | 864 | 865 | 866 | 867 | 868 | 869 | 870 | 871 | 872 | 873 | 874 | 875 | 876 | 877 | 878 | 879 | 880 | 881 | 882 | 883 | 884 | 885 | 886 | 887 | 888 | 889 | 890 | 891 | 892 | 893 | 894 | 895 | 896 | 897 | 898 | 899 | 900 | 901 | 902 | 903 | 904 | 905 | 906 | 907 | 908 | 909 | 910 | 911 | 912 | 913 | 914 | 915 | 916 | 917 | 918 | 919 | 920 | 921 | 922 | 923 | 924 | 925 | 926 | 927 | 928 | 929 | 930 | 931 | 932 | 933 | 934 | 935 | 936 | 937 | 938 | 939 | 940 | 941 | 942 | 943 | 944 | 945 | 946 | 947 | 948 | 949 | 950 | 951 | 952 | 953 | 954 | 955 | 956 | 957 | 958 | 959 | 960 | 961 | 962 | 963 | 964 | 965 | 966 | 967 | 968 | 969 | 970 | 971 | 972 | 973 | 974 | 975 | 976 | 977 | 978 | 979 | 980 | 981 | 982 | 983 | 984 | 985 | 986 | 987 | 988 | 989 | 990 | 991 | 992 | 993 | 994 | 995 | 996 | 997 | 998 | 999 | 1000 | 1001 | 1002 | 1003 | 1004 | 1005 | 1006 | 1007 | 1008 | 1009 | 1010 | 1011 | 1012 | 1013 | 1014 | 1015 | 1016 | 1017 | 1018 | 1019 | 1020 | 1021 | 1022 | 1023 | 1024 | 1025 | 1026 | 1027 | 1028 | 1029 | 1030 | 1031 | 1032 | 1033 | 1034 | 1035 | 1036 | 1037 | 1038 | 1039 | 1040 | 1041 | 1042 | 1043 | 1044 | 1045 | 1046 | 1047 | 1048 | 1049 | 1050 | 1051 | 1052 | 1053 | 1054 | 1055 | 1056 | 1057 | 1058 | 1059 | 1060 | 1061 | 1062 | 1063 | 1064 | 1065 | 1066 | 1067 | 1068 | 1069 | 1070 | 1071 | 1072 | 1073 | 1074 | 1075 | 1076 | 1077 | 1078 | 1079 | 1080 | 1081 | 1082 | 1083 | 1084 | 1085 | 1086 | 1087 | 1088 | 1089 | 1090 | 1091 | 1092 | 1093 | 1094 | 1095 | 1096 | 1097 | 1098 | 1099 | 1100 | 1101 | 1102 | 1103 | 1104 | 1105 | 1106 | 1107 | 1108 | 1109 | 1110 | 1111 | 1112 | 1113 | 1114 | 1115 | 1116 | 1117 | 1118 | 1119 | 1120 | 1121 | 1122 | 1123 | 1124 | 1125 | 1126 | 1127 | 1128 | 1129 | 1130 | 1131 | 1132 | 1133 | 1134 | 1135 | 1136 | 1137 | 1138 | 1139 | 1140 | 1141 | 1142 | 1143 | 1144 | 1145 | 1146 | 1147 | 1148 | 1149 | 1150 | 1151 | 1152 | 1153 | 1154 | 1155 | 1156 | 1157 | 1158 | 1159 | 1160 | 1161 | 1162 | 1163 | 1164 | 1165 | 1166 | 1167 | 1168 | 1169 | 1170 | 1171 | 1172 | 1173 | 1174 | 1175 | 1176 | 1177 | 1178 | 1179 | 1180 | 1181 | 1182 | 1183 | 1184 | 1185 | 1186 | 1187 | 1188 | 1189 | 1190 | 1191 | 1192 | 1193 | 1194 | 1195 | 1196 | 1197 | 1198 | 1199 | 1200 | 1201 | 1202 | 1203 | 1204 | 1205 | 1206 | 1207 | 1208 | 1209 | 1210 | 1211 | 1212 | 1213 | 1214 | 1215 | 1216 | 1217 | 1218 | 1219 | 1220 | 1221 | 1222 | 1223 | 1224 | 1225 | 1226 | 1227 | 1228 | 1229 | 1230 | 1231 | 1232 | 1233 | 1234 | 1235 | 1236 | 1237 | 1238 | 1239 | 1240 | 1241 | 1242 | 1243 | 1244 | 1245 | 1246 | 1247 | 1248 | 1249 | 1250 | 1251 | 1252 | 1253 | 1254 | 1255 | 1256 | 1257 | 1258 | 1259 | 1260 | 1261 | 1262 | 1263 | 1264 | 1265 | 1266 | 1267 | 1268 | 1269 | 1270 | 1271 | 1272 | 1273 | 1274 | 1275 | 1276 | 1277 | 1278 | 1279 | 1280 | 1281 | 1282 | 1283 | 1284 | 1285 | 1286 | 1287 | 1288 | 1289 | 1290 | 1291 | 1292 | 1293 | 1294 | 1295 | 1296 | 1297 | 1298 | 1299 | 1300 | 1301 | 1302 | 1303 | 1304 | 1305 | 1306 | 1307 | 1308 | 1309 | 1310 | 1311 | 1312 | 1313 | 1314 | 1315 | 1316 | 1317 | 1318 | 1319 | 1320 | 1321 | 1322 | 1323 | 1324 | 1325 | 1326 | 1327 | 1328 | 1329 | 1330 | 1331 | 1332 | 1333 | 1334 | 1335 | 1336 | 1337 | 1338 | 1339 | 1340 | 1341 | 1342 | 1343 | 1344 | 1345 | 1346 | 1347 | 1348 | 1349 | 1350 | 1351 | 1352 | 1353 | 1354 | 1355 | 1356 | 1357 | 1358 | 1359 | 1360 | 1361 | 1362 | 1363 | 1364 | 1365 | 1366 | 1367 | 1368 | 1369 | 1370 | 1371 | 1372 | 1373 | 1374 | 1375 | 1376 | 1377 | 1378 | 1379 | 1380 | 1381 | 1382 | 1383 | 1384 | 1385 | 1386 | 1387 | 1388 | 1389 | 1390 | 1391 | 1392 | 1393 | 1394 | 1395 | 1396 | 1397 | 1398 | 1399 | 1400 | 1401 | 1402 | 1403 | 1404 | 1405 | 1406 | 1407 | 1408 | 1409 | 1410 | 1411 | 1412 | 1413 | 1414 | 1415 | 1416 | 1417 | 1418 | 1419 | 1420 | 1421 | 1422 | 1423 | 1424 | 1425 | 1426 | 1427 | 1428 | 1429 | 1430 | 1431 | 1432 | 1433 | 1434 | 1435 | 1436 | 1437 | 1438 | 1439 | 1440 | 1441 | 1442 | 1443 | 1444 | 1445 | 1446 | 1447 | 1448 | 1449 | 1450 | 1451 | 1452 | 1453 | 1454 | 1455 | 1456 | 1457 | 1458 | 1459 | 1460 | 1461 | 1462 | 1463 | 1464 | 1465 | 1466 | 1467 | 1468 | 1469 | 1470 | 1471 | 1472 | 1473 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|